INRIA

UNITÉ DE RECHERCHE
RIA-SOPHIA ANTIPOLIS

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
BP 105
78153 Le Chesnay Cedex
France
Tél. (1) 39 63 55 11

Rapports de Recherche

1992

ème anniversaire

N° 1732

Programme 6
Calcul Scientifique, Modélisation et Logiciels numériques

COMPRESSIBLE FLOW SOLVERS USING UNSTRUCTURED GRIDS

Alain DERVIEUX
Jean-Antoine DESIDERI

Juin 1992

RR-1732

SIMULATION D'ECOULEMENT COMPRESSIBLE UTILISANT DES MAILLAGES NON-STRUCTURES

COMPRESSIBLE FLOW SOLVERS USING UNSTRUCTURED GRIDS

Alain DERVIEUX, Jean-Antoine DESIDERI

ABSTRACT

This document contains a detailed review of works performed at INRIA related to the MUSCL FEM upwind method for unstructured meshes (revised notes of a lecture given at Van Karman Institute).

RESUME

Ce document contient un panorama de travaux effectués à l'INRIA sur la méthode décentrée "MUSCL FEM" applicable aux maillages non-structurés (Notes révisées d'un cours à l'Institut Von Karman).

FOREWORD

Why should we do it simple and cheap when we can do it "unstructured"? The answer is that we think that in the field of compressible flow simulation, a need for solvers handling general geometries and able to perform local refinement in a straight forward manner is of crucial importance in engineering studies. The mythic "complete aircraft configuration" is rather popular ; also consider the geometries of combustion chambers in thermal engines (piston, turbos, rockets).

Because the discrete geometry (triangulation, tetrahedrization) will likely be irregular, we choose to derive the approximation from both recent techniques in upwind-TVD Finite Volume and Finite Element, in order to get an accurate, robust, and parameter-free discrete model (Chapter I).

As a result of this strategy the codes may be very time consuming ; two chapters concentrate on the important issue of efficiency (Chapter II and III).

Chapter IV deals with some recent developments in mesh adaption techniques.

The material covered in these notes compile results obtained by a group of researchers both at INRIA and AMD-BA ; we refer to their work in a "Note" section at the end of each chapter ; we take the opportunity to thank them to their contribution.

CHAPTER I : SPATIAL UPWIND APPROXIMATION

The object of this chapter is to introduce some methods for the spatial discretization over a finite element simplicial mesh (triangles, tetrahedra) of the Euler equations. Ideas both from finite volume and finite element methods are combined ; since the basic ideas were introduced by B. Van Leer in the construction of MUSCL, we shall refer to the schemes presented here as MUSCL-FEM schemes.

1. MATHEMATICAL MODELING

1.1. *Governing equations*

The conservative law form of the Euler equations in the spatially two-dimensional case is given by :

$$W_t + \frac{\partial}{\partial x} F_1(W) + \frac{\partial}{\partial y} F_2(W) = 0 \tag{1}$$

With

$$W = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ E \end{pmatrix} = (W^k)_{k=1,4} \tag{2}$$

and :

$$F_1(W) = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ u(E+p) \end{pmatrix}, \quad F_2(W) = \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ v(E+p) \end{pmatrix} \tag{3}$$

where ρ is the density, $\vec{U} = (u, v)$ is the velocity vector, E is the total energy per unit volume and p is the pressure. We first assume that the fluid satisfies the perfect gas law :

$$p = (\gamma - 1)\,(E - \frac{1}{2}\,\rho|\vec{U}|^2)$$

where γ , ratio of specific heat, to be equal to 1.4 in the air.

Using a symbolic notation for the vectors $(F_i)_{i=1,2}$ in (3),

$$\vec{F}(W) = \begin{pmatrix} F_1(W) \\ F_2(W) \end{pmatrix}$$

then the conservative system (1) becomes :

$$\frac{\partial W}{\partial t} + \vec{\nabla} \cdot \vec{F}(W) = 0.$$

sometimes referred to as the "divergence" form.

1.2. *Boundary conditions*

In the sequel, we consider domains of computation related to external flow around bodies ; in Fig.1 the body is represented by an airfoil which limits the domain of computation by its wall Γ_B ; a second (artificial) boundary is introduced : the farfield boundary denoted by Γ_∞

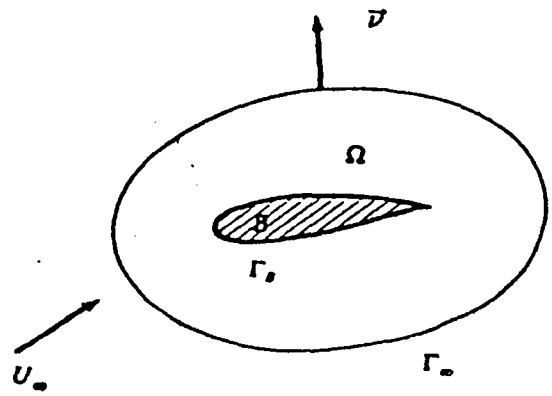


FIGURE 1 :
Boundary of the computational domain Ω

We assume the flow to be uniform at infinity and we prescribe :

$$\rho_\infty = 1 \; ; \; \vec{U}_\infty = \begin{pmatrix} \cos\alpha \\ \sin\alpha \end{pmatrix} \; ; \; p_\infty = \frac{1}{\gamma M_\infty^2}$$

where α is the angle of attack and M_∞ denotes the free-stream Mach number.

On the wall Γ_B we use the following slip condition :

$$\vec{U} \cdot \vec{\nu} = 0.$$

in which $\vec{\nu}$ is the unit vector normal to the wall.

Finally, for (pseudo)unsteady calculations, an initial flow is prescribed :

$$\vec{W}(x,0) = \vec{W}_0(x) \; on \; \Omega$$

2. SPATIAL APPROXIMATIONS IN 2-D

2.1. *Definitions*

We assume that Ω is a polygonal bounded domain of $\mathbb{R}^2$. Let Θ_h a standard [ZIE1] finite element triangulation of Ω and h the maximal length of the sides in Θ_h . We need the following notations :

For every vertex S_i $(i = 1, ..., n_h)$ of Θ_h , we define a cell C_i as follows

- every triangle having S_i as a vertex is subdivided in 6 subtriangles by means of the medians.

- the cell C_i is the union of the resulting subtriangles having S_i as a vertex. The boundary of C_i is denoted by ∂C_i .

We also introduce the following notations

- $K(i)$ is the set of indices of neighboring nodes of S_i.

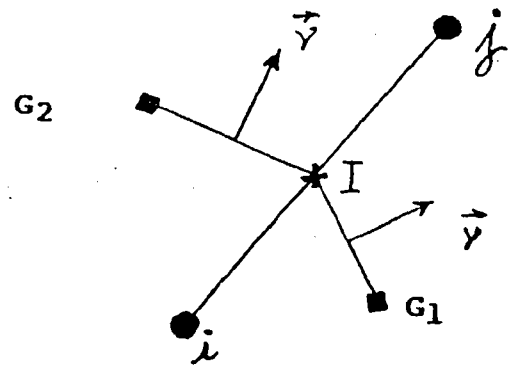
- $\vec{\nu}_i = (\nu_{ix}, \nu_{iy})$ is the unit vector of the outward normal to ∂C_i .

2.2. *First-order accurate approximations*

The two-dimensional extension of the class of three-points upwind first-order accurate 1-D schemes is done as follows [2] : using Green's formula, the Finite-Volume formulation of (1) can be written

$$(4) \qquad area(C_i)\,\frac{W_{h_i}^{n+1} - W_{h_i}^{n}}{\Delta t} + \int_{\partial C_i} \vec{F}(W_h^n)\,.\,\vec{\nu}_i\, d\sigma = 0\,.$$

To compute the integral arising in (4) , we split the surface ∂C_i into panels ∂S_{ij} , $(j \epsilon K(i))$, separating node S_i and node S_j (see Fig.2).



I : midside
G_1, G_2 : triangle centroids

FIGURE 2 :
Portion of a cell boundary. separating node S_i from node S_j .

Then the discrete problem consists of finding, at every new time level t^{n+1} the vector $W^{n+1} = (W_i^{n+1})_i$ where W_i^n holds for the value of W^n at node S_i .

The solution W^{n+1} is defined by the discrete system :

$$
\begin{aligned}
& area(C_i)\ \frac{W_i^{n+1} - W_i^n}{\Delta t} \\
& \sum_{j \in K(i)} \int_{\partial C_{ij}} \vec{F}(W^n) \cdot \vec{\nu}_i \, d\sigma \\
& + \int_{\partial C_i \cap \Gamma_B} \vec{F}(\bar{W}^n) \cdot \vec{\nu} \, d\sigma \\
& + \int_{\partial C_i \cap \Gamma_\infty} \vec{F}(\bar{W}^n) \cdot \vec{\nu} \, d\sigma = 0
\end{aligned}
$$

the notation $\bar{W}^n$ will be explained in the sequel ; we now specify the computation of the integral over ∂S_{ij} ; actually, the evaluation of this term corresponds to the one-dimensional calculation of the flux along the direction S_iS_j , that will be computed from the following metric coefficients:

$$
\vec{\eta}^{ij} = \int_{\partial C_{ij}} \vec{\nu}_i \, d\sigma
$$

For upwinding, we introduce a numerical flux function ; the approximate system

becomes :

$$
\begin{aligned}
& area(C_i)\ \frac{W_i^{n+1} - W_i^n}{\Delta t} \\
& + \sum_{j \in K(i)} \Phi(W_i^n, W_j^n, \eta^{ij}) \\
& + \int_{\partial C_i \cap \Gamma_B} \vec{F}(\bar{W}_h^n) \cdot \vec{\nu}\ d\sigma \\
& + \int_{\partial C_i \cap \Gamma_\infty} \vec{F}(\bar{W}_\infty^n) \cdot \vec{\nu}\ d\sigma\ = 0
\end{aligned}
\tag{5}
$$

where $\Phi(U, V, \vec{\eta})$ denotes a numerical flux splitting : we choose Osher's splitting [19] for its good stability properties when applied to high Mach number flow simulation. In addition, it results in a differentiable flux function, which can be crucial in correctly capturing stagnation zones and sonic points, where the Jacobian matrices are singular. This flux function is given in short by :

$$
\begin{aligned}
F_{ij} &= \eta_x^{ij}\ F_1 + \eta_y^{ij}\ F_2 \\
\Phi^{OSHER}(U, V, \vec{\eta}^{ij}) &= \frac{F_{ij}(U) + F_{ij}(V)}{2} - \frac{1}{2} \int_U^V |F_{ij}'(W)| dW
\end{aligned}
$$

The integration from U to V is performed successively along three paths defined by the Riemann invariants ; these paths are uniquely defined if we except symmetry ; the signs of eigenvalues are taken into account for exact integration.

Integrals over Γ_B and Γ_∞ in (5) involve the physical boundary conditions, that are taken into account through the vector $\bar{W}^n$: this vector is computed from quantities depending on the node values W^n and quantities derived from the physical boundary conditions

For a wall boundary integral, the vector $\bar{W}^n$ is defined as satisfying the slip condition and thus the integral over Γ_B is written :

$$
\int_{\partial C_i \cap \Gamma_B} \vec{F}(\bar{W}^n) \cdot \vec{\nu}\ d\sigma\ + \int_{\partial C_i \cap \Gamma_B} \begin{pmatrix} 0 \\ \bar{p}\ \nu_x \\ \bar{p}\ \nu_y \\ 0 \end{pmatrix} d\sigma\ ,
$$

where $\bar{p}$ is equal to the node pressure $p(W^n)$.

With this procedure, the slip condition is applied in a weak variational way.

For inflow and outflow boundary integrals, we have to select a precise set of exterior datas, depending on the flow regime and the velocity direction ; for this purpose, a plus-minus splitting is applied between exterior datas and interior (=node) values. More precisely, the integral is evaluated as follows :

$$(6) \qquad \int_{\partial C_i \cap \Gamma_\infty} \vec{F}(\bar{W}^n) \cdot \vec{\nu} \; d\sigma \; = \; P^+_{i\infty}(W^n_i) W^n_i \; + \; P^-_{i\infty}(W^n_i) W_\infty$$

where

$$(7) \qquad P_{i\infty}(U) \; = \; \vec{F}'(U) \int_{\partial C_i \cap \Gamma_\infty} \vec{\nu} \; d\sigma$$

and where P^+ and P^- are derived from the diagonalization of P :

$$(8) \quad P \; = \; T\Lambda T^{-1} \; , \; \Lambda \; \textit{diagonal} \; ; \; P^+ \; = \; T\Lambda^+ T^{-1} \; ; \; P^- \; = \; T\Lambda^- T^{-1} \; .$$

2.3. *Second-order accurate approximations*

With constant-by-cell dependent variables, the above numerical split-flux integration will result in schemes which are only first-order accurate. Higher-order schemes without TVD devices do provide very accurate solutions in smooth regions, but produce oscillatory results rear discontinuities.

The main ingredients in constructing a (non-oscillatory) second-order accurate approximation include :
(i) the above first-order quasi-monotone scheme with upwind flux function ;
(ii) a second-order variant using linear interpolations ;
(iii) a slope-limiting procedure to reduce the oscillations in the solution.

This approach intends to extend Van Leer's MUSCL method [29] to non-structured meshes.

2.3.1. Hermitian limiter

This kind of slope limiter directly acts in the computation of approximate node gradients of flow variables ; the employed flow variables are the density, the velocity components and the pressure :

$$\hat{W} \; = \; (\rho, u, v, p)$$

The approximate x-derivative, for example, is computed as the value of least modulus among the Galerkin x-derivatives in the neighboring triangles if they are all of the same sign, and as zero otherwise :

$$\frac{\partial \hat{W}^{lim}}{\partial x}(i) = \min_{T \text{ neighbor of } i} \text{mod} \left(\frac{\partial \hat{W}}{\partial x}|_T \right). \tag{9}$$

This is applied in the same way for boundary nodes. These quantities are then used for the interpolation of the flow variables :

$$\hat{W}_{ij} = \hat{W}_i + \frac{1}{2} \begin{pmatrix} \frac{\partial \hat{W}}{\partial x}(i) \\ \frac{\partial \hat{W}}{\partial y}(i) \end{pmatrix} . \vec{ij}$$

$$\hat{W}_{ij} = \hat{W}_j + \frac{1}{2} \begin{pmatrix} \frac{\partial \hat{W}}{\partial x}(j) \\ \frac{\partial \hat{W}}{\partial y}(j) \end{pmatrix} . \vec{ji}$$

From this conservative variables W_{ij} and W_{ji} are derived and used in the split-flux ; this results in the following discrete equation :

$$\begin{aligned} &\frac{area(C_i)}{\Delta t}(W_i^{n+1} - W_i^n) + \sum_{j \in K(i)} H_{ij}^{(2)} \\ &+ \int_{\partial C_i \cap \Gamma_B} F(\bar{W}^n) . \nu d\sigma + \int_{\partial C_i \cap \Gamma_\infty} F(\bar{W}^n) . \nu d\sigma = 0, \end{aligned} \tag{10}$$

with

$$H_{ij}^{(2)} = \Phi(W_{ij}, W_{ji}, \vec{\eta}^{ij})$$

2.3.2. Upwind-element formulation

Another way to reach second-order accuracy, is the following : for every segment S_iS_j, we can define uniquely two elements T_{ij} and T_{ji} to be such that

$$\begin{aligned} S_i + \lambda \vec{S_jS_i} \in T_{ij} , \\ S_j + \lambda \vec{S_iS_j} \in T_{ji} , \end{aligned} \tag{11}$$

provided λ is a small enough positive number.

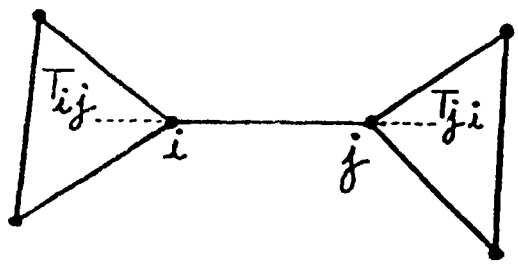


FIGURE 3
Upwind triangles for a given side ij

On these two "upwind triangles" we compute the (Galerkin) gradients $\vec{\nabla}\hat{W}|_{T_{ij}}$ of $\hat{W}$. We then derive the following interpolations :

$$\hat{W}_{ij}^u = \hat{W}_i + \frac{1}{2}\vec{\nabla}\hat{W}|_{T_{ij}} \cdot \vec{ij}$$
$$\hat{W}_{ij}^u = \hat{W}_j + \frac{1}{2}\vec{\nabla}\hat{W}|_{T_{ij}} \cdot \vec{ji}$$

Then the use of W_{ij}^u and W_{ji}^u in the flux splitting will result in a fully upwind second-order accurate scheme.

In chapter II, it is shown that there is some avantage in employing s "half-upwind"scheme. This scheme is obtained by averaging this previous estimates with the variation of $\hat{W}$ over segment ij :

$$\hat{W}_{ij}^h = \hat{W}_i + \frac{1}{4}\vec{\nabla}\hat{W}|_{T_{ij}} \cdot \vec{ij} + \frac{1}{4}(\hat{W}_j - \hat{W}_i)$$
$$\hat{W}_{ji}^h = \hat{W}_j + \frac{1}{4}\vec{\nabla}\hat{W}|_{T_{ij}} \cdot \vec{ji} + \frac{1}{4}(\hat{W}_i - \hat{W}_j)$$

The slope-limited half-upwind scheme that we propose is obtained by adding a limiting process that applies along the directions of the segment ij :

$$\hat{W}_{ij} = \hat{W}_i - \frac{1}{2}Ave[2(\hat{W}_i - \hat{W}_{ij}^u), \hat{W}_i - \hat{W}_j]$$
$$\hat{W}_{ji} = \hat{W}_j - \frac{1}{2}Ave[2(\hat{W}_j - \hat{W}_{ij}^u), \hat{W}_j - \hat{W}_i]$$

with, following [LEE1] :

$$Ave(a,b) = \frac{a(b^2 + \epsilon) + b(a^2 + \epsilon)}{a^2 + b^2 + \epsilon}$$

where $\epsilon > 0$ prevents the zero divide. The extension of this scheme to boundary nodes is not well solved.

One advantage of this method is that, since the limitation acts in one direction, we can replace $\hat{W}$ by characteristic values along this direction.

3. SPATIAL APPROXIMATION IN 3-D

Every concept of the 2-D construction is easily extended to 3-D ; let us make more precise some geometrical points :

The computational domain Ω is assumed to be a polyhedral bounded subdomain of $\mathbb{R}^3$, and Θ_h is a standard finite element tetrahedrization, with h as maximal length of the edges in Θ_h.

A dual finite volume partition is derived from the construction of median planes, that is, for every vertex S_i of Θ_h, we define a Cell or finite volume C_i "around S_i" as follows

- every tetrahedron having S_i as a vertex is subdivided in 24 subtetrahedra by means of planes containing an edge and the middle of the opposite edge ; then the cell C_i is the union of subtetrahedra having S_i as a vertex.

In particular, the boundary ∂C_i of C_i is the union of the $\partial C_{ij} = \partial C_i \cap \partial C_j$ that can be defined as the union of triangles such that (Fig. 4)
- one vertex is the middle of the edge S_iS_j
- one vertex is the centroid of the tetrahedron T having S_iS_j as an edge
- one vertex is the centroid of a (triangular) face of T having S_iS_j as a side.

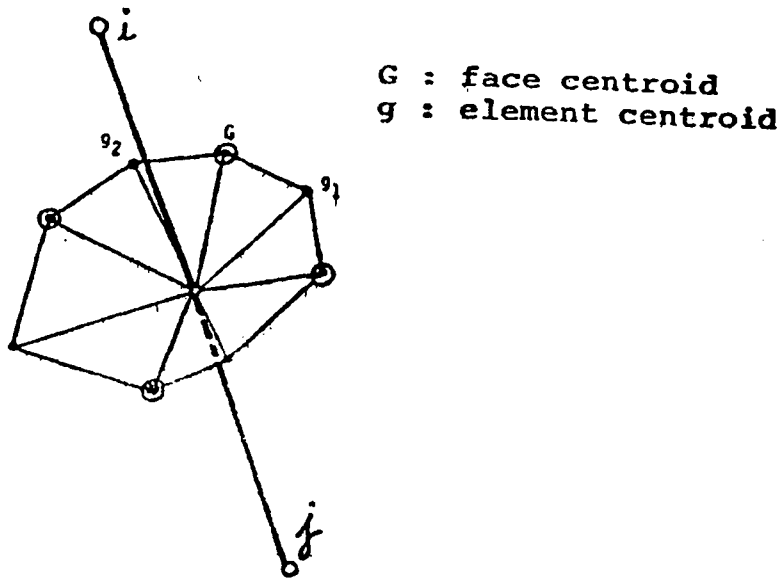


FIGURE 4 :
Sketch of $\partial C_{ij} = \partial C_i \cap \partial C_j$

Lastly, the upwind-element formulation will rely on the same definition (11), sketched now by Fig. 5.

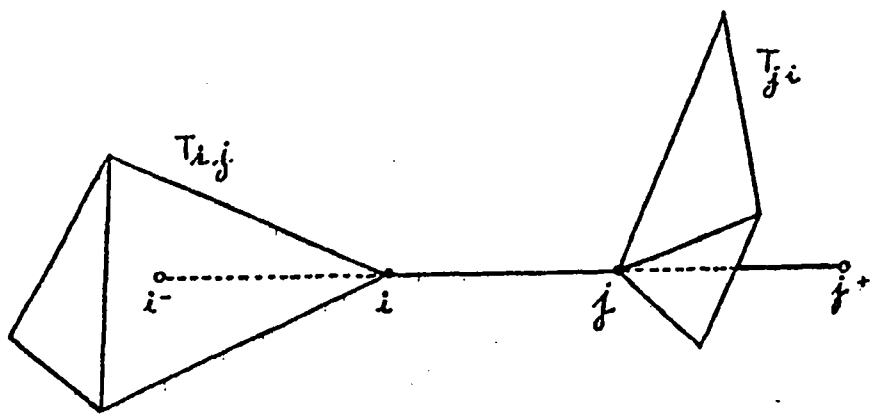


FIGURE 5 :
Upwind tetrahedra. T_{ij} *and* T_{ji} for a given edge ij

4. NOTES

First-order schemes with triangles were derived by TABATA (scalar models) [BAB] and Vijayasundaram [VIJ]. The first MUSCL-FEM formulation is due to Fezoui [FEZ1] ; the construction of robust slope-limiters, able to approximate stably high Mach number flows around a cylinder seems to necessitate devices as heavy as the slope limiters presented here [STO3]. Of course the MUSCL-FEM is not the only one approach for unstructured incompressible models ; we refer to Taylor-Galerkin ones, see [SEL] [LOH2] [DER1], to the pure Galerkin one [MAV] and to the Petrov-Galerkin one [HUG].

CHAPTER II : IMPLICIT UNFACTORED DELTA SCHEMES

Implicit unfactored delta schemes have gained a new interest for the construction of fast steady-Euler solvers [STO1, LEE3, CHA, MC, STO3]. They are, when combined with a relaxation iteration, rather easily derived for unstructured grids and in this context have been studied quite early (see [STO1] for Galerkin calculations). We present in this section a general procedure to derive such solvers from an upwind first-order accurate approximation, acting as a preconditioner, in combination with the MUSCL-FEM second-order accurate upwind approximation.

1. BASIC FEATURES OF UNFACTORED SCHEMES

1.1. *Model Problems and Model Differencing Schemes*

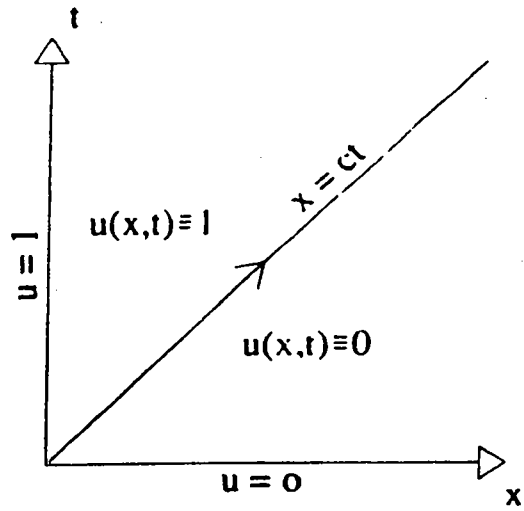
In one dimension, the Euler equations can be written in the following quasi-linear form:

$$W_t + AW_x = 0 \tag{1}$$

in which $A = A(W)$ is the usual 3×3 Jacobian matrix, that can be expressed and diagonalized explicitly. To allow a **linear analysis** of numerical schemes, one may **construct a model hyperbolic test equation by setting A to a constant matrix** ; the above system reduces to a set of "**convection equations**", and an appropriate model is given by the following "**quarter-plane problem**":

$$\begin{cases} u_t + cu_x = 0 & (c > 0;\ t > 0,\ x > 0) \\ u(x,0) = 0 & (x > 0) \\ u(0,t) = 1 & (t > 0) \end{cases} \tag{2}$$

This is a purely convective problem, in which information travels without dissipation along characteristics, $x - ct$ = constant. Nevertheless, the exact solution over the spatial interval $[0, X]$ (for some fixed X), is stationary for $t \geq \frac{X}{c}$ (and uniform): $u(x,t) \equiv 1$. The process of convergence (to steady state) is therefore distinct from that of a dissipative phenomenon (e.g. heat equation). However, in the discrete models, dissipation does exist in the form of "**artificial dissipation**".



For the model problem, various differencing schemes may be employed to represent the spatial derivative u_x (multiplied by Δx): central differencing, δ^c_x, 1st-order backward differencing, $\delta^u_{x,1}$, 2nd-order backward differencing, $\delta^u_{x,2}$. *. The matrix analogs of these operators are defined in details in Appendix I in the periodic case, and in the case where some appropriate non-periodic boundary conditions are applied. The corresponding sets of eigenvalues are described on the figure. It should be stressed that the two cases although represented by matrices that differ from one another by a few elements only, are very disctinct in nature. The periodic model is adequate to carry out a Fourier analysis, yielding L2-stability conditions, and phase-error and wavespeed evaluations. However, the non-periodic model is more appropriate for iterative convergence rate estimations.

The following general statements can be made:
(1) All the **eigenvalues** λ_m of any acceptable differencing scheme **fall on the same half-plane**:

$$\forall m. \quad Re(\lambda_m) \geq 0 \tag{3}$$

(2) The **central-difference** operator can be **diagonalized**. The eigenvalues are (or are close to) **purely imaginary** which indicates the absence of artificial viscosity. (In the non-periodic case, all but one eigenvalues can be shown to satisfy: $Re(\lambda_m) = O(1/N^2)$.)
(3) In the non-periodic case, **upwind schemes** are represented by **defective (i.e. non diagonalizable) triangular** matrices with multiple eigenvalues, whose real parts are of order 1.

---

* Backward differences are considered in the definition of upwind schemes since $c > 0$ has been assumed

The first of these statements is a "well-posedness requirement" for the spatial-differencing scheme, since it is equivalent to the stability of the time-continuous, space-discrete equation

$$U_t + D_h U = 0 \tag{4}$$

in which $U = (u_1(t), u_2(t), \cdots, u_N(t))^T$ is the space-discretized unknown vector, and $D_h U$ is the employed discrete approximation of cU_x ($D_h = \frac{c}{\Delta x}\delta_x^c$, or $\frac{c}{\Delta x}\delta_{x,1}^u$, or $\frac{c}{\Delta x}\delta_{x,2}^u$, $\cdots$), since the solution of (4) is given by:

$$U(t) = \sum_m c_m e^{-\frac{c\lambda_m t}{\Delta x}} V_m \tag{5}$$

in which V_m is the mth eigenvector of D_h, λ_m the associated eigenvalue, and c_m the component along V_m of the initial solution. Note that the stability of the space differencing is usually a necessary stability condition for the time-dicrete scheme, but certainly not a sufficient one, since the time integration of (4) has been performed exactly, and involves non timestep limitation.

We now turn our attention to upwind schemes. The real parts of their eigenvalues are of order 1. This corresponds to the important amount of artificial dissipation inherent to this type of approximation. In view of (5) it appears that this implies that in the convergence to steady state along with convection of errors away across and away from the domain, dissipation at finite rate plays a role. The fact that these operators are represented in the model problem by defective (triangular) matrices has important consequences on the nature and properties of the iterative convergence (see Sections 1.4 and 1.5).

Finally, observe that in the case of the Euler equations, upwind schemes for equation (1) are frequently constructed via flux-splitting which isolates the contributions from the positive and the negative eigenvalues of the Jacobian matrix A (represented in the model problem by the single constant $c > 0$) prior to applying a backward or forward type differencing scheme.

1.2. *Construction of Implicit Schemes - Motivations*

We now turn to the time-discretization method, sometimes referred to as the "solver". For the solution of steady problems, implicit schemes are attractive since they are not limited by the C.F.L. stability condition, and therefore allow rapid convergence to steady state when large timesteps are employed. Here we concentrate on the "**linearized backward Euler scheme**" also known as the "**fully implicit method**". When applied to the Euler equations, it can be written in the following "**delta form**":

$$M_h \left(W^{n+1} - W^n\right) = -\Delta t D_h W^n \tag{6}$$

where Δt is the timestep and the operator M_h is defined by

$$M_h = Id + \Delta t\,(D_h W^n)_w \tag{7}$$

in which the subscript w indicates that the Jacobian is formed. (For this classical derivation see for example [BEA]). To evaluate the stability of this method we consider again the linear hyperbolic model, for which D_h and M_h can be thought as matrices constant during the iteration and satisfying

$$M_h = Id + \Delta t D_h \tag{8}$$

so that, an amplification matrix $G_{\Delta t}$ can be defined by

$$W^{n+1} = G_{\Delta t} W^n + b \tag{9}$$

in which b is a constant vector containing prescribed boundary terms, and turns out to be:

$$G_{\Delta t} = Id - \Delta t M_h^{-1} D_h = Id - \left(Id + (\Delta t D_h)^{-1}\right)^{-1} \tag{10}$$

Thus if the eigenvalues of D_h are denoted as previously by λ_m $(m = 1, 2, \cdots, N)$, those of $G_{\Delta t}$ are given by

$$g_m(\Delta t) = 1 - \frac{1}{1 + \frac{1}{z_m}} = \frac{1}{z_m + 1} \tag{11}$$

where $z_m = \lambda_m \Delta t$. Since for all m,

$$Re(z_m) \geq 0 \tag{12}$$

as established in the previous section. it follows that for all Δt

$$|g_m(\Delta t)| \leq 1 \tag{13}$$

thus proving that the method is "**unconditionally stable for the associated linear hyperbolic problem**". Furthermore,

$$\lim_{\Delta t \to \infty} g_m(\Delta t) = 0 \tag{14}$$

Note that this favorable property is not true in 2-D for factored schemes whose eigenvalues are of the form (see Appendix 2 in [STE2])

$$g_{m,k}(\Delta t) = 1 - \frac{(\lambda_m + \mu_k)\Delta t}{(1 + \lambda_m \Delta t)(1 + \mu_k \Delta t)} \tag{15}$$

so that

$$\lim_{\Delta t \to \infty} g_{m,k}(\Delta t) = 1 \qquad \text{(2-D factored schemes)} \tag{16}$$

Of course these ideal results, valid for a linear model, may not entirely extend to the nonlinear case. However, the Euler implicit method is stable for values of Δt that are not "**not limited by the C.F.L. condition**" which restricts the usual explicit schemes, and becomes "**more dissipative with larger timesteps**" while the steady-state solution which is independent of Δt is only determined by the differencing operator D_h appearing explicitly on the right-hand side.

Note that if one lets $\Delta t \longrightarrow \infty$ and defines $\Phi(W) = D_h w$, then (6) becomes

$$\Phi_w(W^n)\,(W^{n+1} - W^n) = -\Phi\,(W^n) \tag{17}$$

In this formulation, we recognize "**Newton's method**" whose convergence is **quadratic** [JMO1]. (For a linear problem, the amplification matrix G_∞ is then equal to the null matrix and the process converges in one iteration). This confirms that being able to use stably very large time-steps is a highly desirable feature for the solver. Again this property falls for factored schemes.

We now examine the algorithmic standpoint. The application of the algorithm defined in (6) is performed in three steps:
(1) "**Physical phase**":
computation of the "right-hand side vector" $R = -\Delta t D_h W^n$
(2) "**Mathematical phase**":
solution of the system $M_h \Delta W^n = R$, in which the unknown is the vector ΔW^n;
(3) "**Update**":

$$W^{n+1} = W^n + \Delta W^n \tag{18}$$

The implicit mathematical phase preconditions the system in a way that enhances the stability of the method, but has no effect on steady-state accuracy. The physical phase however, which is completed solely after proper boundary conditions are enforced, defines alone the converged solution. Therefore we require that the operator D_h be at least **second-order accurate** in regions where the solution is smooth. This is achieved either by a **central differencing scheme** [BEA, STE2, STO1] or the-like, or instead by a **second-order upwind scheme**. This alternative has gained some popularity in recent papers [STE3, LEE3, THO, MC, STO3], because it yields schemes having (or almost having) certain desirable monotonicity properties and thus producing more physically relevant solutions near discontinuities, that is, optimally, accurate and oscillation-free solutions. Another alternative is to combine

(linearly) a central discretization with an upwind discretization. In any case the requirement of obtaining a second-order accurate space discretization is generally not very difficult to meet because the step is explicit; moreover, in many centered schemes, no Jacobians need be calculated but only linear combinations of the flux vectors f and g computed at various nodes; that is the approximation $D_h W^n$ is computed without having to explicitly evaluate the operator D_h. In contrast, the mathematical phase is far more complicated to realize, and this for several reasons:

- It is imperative to evaluate the operator M_h itself. If splitting is not used, the constituent blocks of M_h are linear combinations of the 4x4 Jacobians (in 2-D), that involve 4 times more (known) functions to compute than in the flux vectors themselves. If splitting is applied, the true Jacobian $(D_h W^n)_w$ may be difficult to express in closed-form, while an approximate linearization may already be quite involved computationally. In addition, the precise linearization of the boundary procedure in a 2nd-order scheme is also difficult.
- The system $M_h \Delta W^n = b$ need be solved. This is always a computationally difficult task when the system is large, particularly when the mesh is not regular in structure and smooth, or when the discretization is very sophisticated and complex. yielding **a large and ill-conditioned matrix (stiffness)**
- The inversion process necessitates the storage of the constituent blocks of the matrix M_h. When this matrix is too large, the limit of the computer's storage capability is attained and there is no alternative to overwriting certain blocks and reevaluating them everytime they reappear in a subsequent calculation. Thus some of the Jacobians are evaluated more than once, possibly the splitting of the flux vectors also, and more work is required than would be expected solely by inspection of the mathematical equations.

For all of these reasons, many authors are using simplified versions of the implicit method, in which the Euler equations are approximated only to first-order in the mathematical phase, while the physical phase remains the same. This **reduces the bandwidth** of the matrix M_h and thus significantly lessens the amount of computation done in the mathematical phase. In addition there is an important advantage in using the first-order scheme implicitly: in doing so, and at least for the model problem, the matrix system to be solved at each iteration is **diagonally dominant**. Therefore the inversion can be performed by **relaxation**, either by the **Gauss-Seidel** iteration [CHA, FEZ3], or by the **Point-Jacobi** iteration [ANG5, STEV2] which has regained interest in **vectorized** computations.

However with a first-order implicit "preconditioner", a slight inconsistency in the formulation is introduced, and the efficiency of the method at large timesteps can no longer be that of Newton's method. The rate of convergence to steady state is evaluated in the next section for a class of implicit schemes in which a parameter β $(0 < \beta < 1)$ controls the degree of upwinding introduced in the 2nd-order differencing scheme of the explict phase. The limiting cases ($\beta = 0$, central

differencing, and $\beta = 1$, fully upwind scheme) are shown to be pathological in iterative convergence in the subsequent section.

1.3. *Diagonalization of Implicit Upwind Schemes - Rate of Convergence*

In this section, we analyze the iterative convergence of the implicit delta scheme applied to the model problem in the case where a **second-order** difference operator of **adjustable upwinding** is employed in the explicit phase,

$$D_h = \frac{c}{\Delta x}\delta^{\beta}_{x,2} \tag{19}$$

where $\delta^{\beta}_{x,2}$ combines the fully-upwind scheme with the central differencing scheme,

$$\delta^{\beta}_{x,2} = \beta\delta^{u}_{x,2} + (1-\beta)\delta^{c}_{x} \tag{20}$$

$(0 \leq \beta \leq 1)$, and a **first-order** upwind scheme is applied in the implicit phase:

$$M_h = Id + \frac{c\Delta t}{\Delta x}\delta^{u}_{x,1} \tag{21}$$

Thus the iteration is shortly defined by

$$\left(Id + \frac{c\Delta t}{\Delta x}\delta^{u}_{x,1}\right)\left(U^{n+1} - U^{n}\right) = -\frac{c\Delta t}{\Delta x}\delta^{\beta}_{x,2}U^{n} \tag{22}$$

This allows us to again define the amplification matrix, $G_{\Delta t}$, by

$$U^{n+1} = G_{\Delta t}U^{n} + b \tag{23}$$

Consequently, when $\Delta t \longrightarrow \infty$, the amplification matrix $G_{\Delta t}$ approaches

$$G_{\infty} = Id - \left(\delta^{u}_{x,1}\right)^{-1}\delta^{\beta}_{x,2} \tag{24}$$

Thus although the timestep is infinite and the problem linear, the amplification matrix is nonzero, the iteration is not equivalent to Newton's method, and the asymptotic convergence can at best be linear. Actually, this is the case only when the values $\beta = 0$ or 1 are excluded.

It should be noted that for the linear model problem, this identifies to a particular application of the so-called "**Defect-Correction Method**" [BOH] defined

as follows: Let $\Phi_1(u)$ and $\Phi_2(u)$ be first-order and second-order discrete approximations of the steady-state equation; then solve first the equation $\Phi_1(u^0) = 0$ for u^0; then, for $n = 0, 1, \cdots$, solve

$$\Phi_1(U^{n+1}) = \Phi_1(U^n) - \Phi_2(U^n) \tag{25}$$

In this way, second-order approximations are evaluated only to form right-hand sides. Inversions only involve the simpler first-order approximation scheme. In our case,

$$\Phi_1 = \frac{c}{\Delta x}\delta^u_{x,1}, \qquad \Phi_2 = \frac{c}{\Delta x}\delta^\beta_{x,2} \tag{26}$$

and the identification is obvious.

For $0 < \beta < 1$, the eigenvalues are distinct [DES2]:

$$\begin{cases} \lambda_0 = 0 \\ \lambda_m = \dfrac{1}{2} - \beta + i\sqrt{\beta(1-\beta)}\cos\dfrac{m\pi}{N} \qquad (m = 1, 2, \cdots, N-1) \end{cases} \tag{27}$$

A diagram of these eigenvalues in which β is a parameter is given by Figure 1. The eigenvectors can be expressed explicitly, each one being a simple function of the corresponding eigenvalue; the matrix G_∞ is diagonalizable * and the convergence is (immediately) dissipative, at a rate slightly more rapid than that of the sequence 2^{-n}, since the spectral radius is given by:

$$\begin{aligned} \rho &= |\lambda_1| \\ &= \frac{1}{2}\sqrt{1 - 4\beta(1-\beta)\sin^2\frac{\pi}{N}} < \frac{1}{2}, \quad \text{and} \approx \frac{1}{2} \end{aligned} \tag{28}$$

This result is remarquable since it implies that the iterative convergence rate is essentially **independent of meshsize**. The best separation of the eigenvalues and the best condition system of eigenvectors are realized by the **half-fully upwind** scheme ($\beta = \frac{1}{2}$), presumably the most robust scheme. Also of interest the $\frac{1}{3}$-**fully upwind scheme** which is spatially **third-order accurate**.

These results can be somewhat extended to two dimensions, in the case of the following model equation

$$u_t + au_x + bu_y = 0 \tag{29}$$

Assuming a mesh of $N_x \times N_y$ gridpoints, one denotes the Kronecker product (or tensor product) by the symbol $\otimes$ and the Kronecker sum by the symbol $\oplus$, and one recalls that if A and B are matrices of dimensions respectively $N_x \times N_x$ and $N_y \times N_y$,

---

* For $\beta = \frac{1}{2}$ we are discarding the case where N is even, for which the eigenvalue $\lambda = 0$ is double and the matrix defective; however, this has no severe consequence on the convergence rate since only one eigenvector is missing, as it will be explained in the next section.

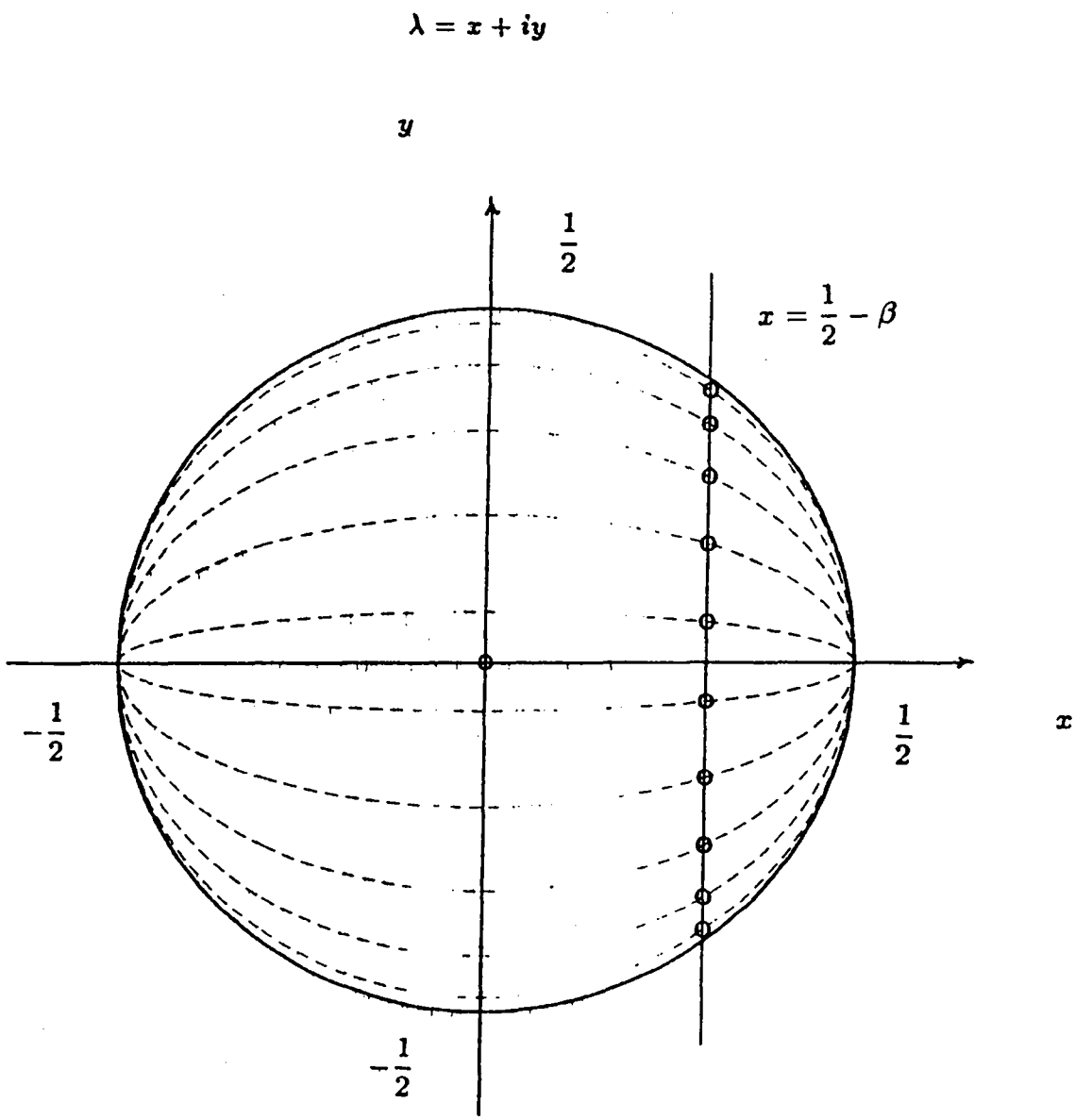


Figure 1 : Eigenvalues of the Euler Implicit Method in Δ-form for the 1-D non-periodic model problem with explicitly, 2nd-order β-fully upwind scheme, implicitly, first-order upwind scheme, and infinite timestep

then the matrices $C = A \otimes B$ and $D = A \oplus B$ are of dimension $N_x N_y \times N_x N_y$ and are defined by:

$$C = A \otimes B = \begin{pmatrix} a_{1,1}B & a_{1,2}B & \cdots & a_{1,N_x}B \\ a_{2,1}B & a_{2,2}B & \cdots & a_{2,N_x}B \\ \vdots & \vdots & & \vdots \\ a_{N_x,1}B & a_{N_x,2}B & \cdots & a_{N_x,N_x}B \end{pmatrix} \tag{30}$$

and

$$D = A \oplus B = A \otimes I_y + I_x \otimes B \tag{31}$$

in which I_x and I_y denote the identity matrices of dimensions $N_x \times N_x$ and $N_y \times N_y$ respectively. With these notations, the expression of the limiting amplification matrix G_∞ becomes:

$$G_\infty = Id - \left(\delta^u_{xy,1}\right)^{-1} \delta^\beta_{xy,2} \tag{32}$$

provided one defines the first-order and second-order difference operators as the following Kronecker sums, which depend on the wavespeeds a and b:

$$\begin{aligned} \delta^u_{xy,1} &= \left(a\delta^u_{x,1}\right) \oplus \left(b\delta^u_{y,1}\right) \\ \delta^\beta_{xy,2} &= \left(a\delta^\beta_{x,2}\right) \oplus \left(b\delta^\beta_{y,2}\right) \end{aligned} \tag{33}$$

If $N_x = N_y$, it can easily be shown that if λ is an eigenvalue of the 1-D model problem and u an associated eigenvector, then $u \otimes u$ is an eigenvector of the 2-D model problem associated with the same eigenvalue λ, independently of the values of the wavespeeds a and b. Further, if we assume that in 2-D the critical eigenvector associated with the eigenvalue of modulus ρ is the 2-D highest-frequency mode, that is the tensor product of 1-D highest frequency modes, then the spectral radius ρ is the same in 2-D as in 1-D, that is $\rho \approx \frac{1}{2}$ independently of the wavespeeds a and b. This result is confirmed by an experiment in [DES1, Table 3]. If $N_x \neq N_y$, the same is true in the limit $N_x,\ N_y \longrightarrow \infty$ since the critical eigenvalue $\lambda \longrightarrow \frac{1}{2} - \beta + i\sqrt{\beta(1-\beta)}$ is then common to x and y 1-D operators. Consequently, the following conjecture is made:

Conjecture: When $\Delta t \longrightarrow \infty$, the linearized implicit Euler method based on the second-order partially upwind scheme explicitly ($0 < \beta < 1$), and the first-order upwind scheme implicitly, converges iteratively, in 1-D as in 2-D, at the rate of the sequence 2^{-n}, where n is the iteration number, independently of the mesh.

1.4. *Pathological Schemes*

We begin this section by examining the application of the simple explicit first-order upwind scheme to (2):

$$u_j^{n+1} = u_j^n - c\Delta t \frac{u_j^n - u_{j-1}^n}{\Delta x} \tag{34}$$

with a Courant number,

$$\nu = \frac{c\Delta t}{\Delta x} \tag{35}$$

set equal to 1, the method reduces to the method of characteristics (Which is exact):

$$u_j^{n+1} = u_{j-1}^n \tag{36}$$

If we let

$$U^n = \begin{pmatrix} u_1^n \\ u_2^n \\ \vdots \\ u_N^n \end{pmatrix} \tag{37}$$

where N denotes the number of mesh-intervals ($\Delta x = \frac{X}{N}$).

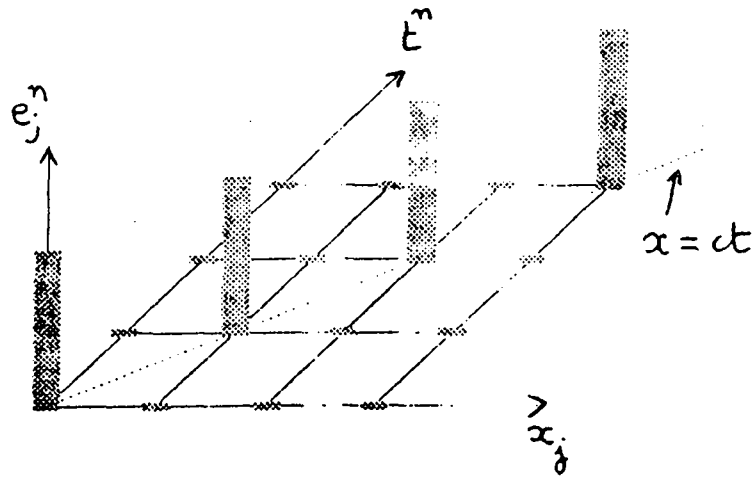
$$U^{n+1} = GU^n + b \tag{38}$$

where

$$G = \begin{pmatrix} 0 & & & & \\ 1 & 0 & & & \\ & 1 & 0 & & \\ & & \ddots & \ddots & \\ & & & 1 & 0 \end{pmatrix} \quad , \quad b = \begin{pmatrix} u_0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \tag{39}$$

The amplification matrix G is defective, and although the spectral radius $\rho = 0$, the steady-state solution is not found in 1 iteration only, but in $N - 1$. The matrix G is a Jordan block of order $N \times N$; $N - 1$ eigenvectors are missing, and the convergence process begins with a phase of **transfer** of the components of the "error-vector", $u^n - u^\infty$ (u^∞ denotes the steady-state solution), **from a singular vector to the next**. This phase extends over $N - 1$ iterations, that is according to the analysis of Appendix II, the ratio of the number of missing eigenvectors, $N - 1$, to $1 - \rho$, where ρ is the spectral radius (here $\rho = 0$). Then only, after the error content is "flipped" into the only true eigenvector, the dissipative phase begins. Here, this phase reduces to immediate annihilation (in 1 iteration) since $\rho = 0$. In this case, the convective phase corresponds to an exact integration of the P.D.E. (With exact finite wave-propagation speed) and is no surprise.

The sketch below illustrates the convection of an error signal initially concentrated at the left-boundary point by the simple explicit method when the Courant number is equal to unity. The signal travels one meshinterval at each iteration.



However, this pattern of convergence that exhibits a phase of significant extent during which the norm of the residual (expressed in the basis of the singular vectors) is not reduced, can be observed anytime the iteration is defective and the number of missing eigenvectors is large. By analogy with the convergence of the simple explicit method we refer to this phase as one of **"pseudo-convection"**.

This is the case in particular for the implicit methods under study when the timestep is infinite and the upwinding parameter β is set to either limit 0 or 1. To see this, return to Figure 1. In either limit, the spectral radius is equal to $\frac{1}{2}$; however, $N-1$ eigenvalues are identical and the corresponding eigenvectors that can be expressed in closed-form as $v^m = \chi(\lambda_m)$, for the same known vector-valued function $\chi(\lambda)$, coalesce. Hence the matrix is defective, and the number of missing eigenvectors, $N-2$, is large. As a result, the pseudo-convection phase extends over a number of iterations equivalent, for N large, to $\frac{N}{1-\rho} = 2N$. Here, the **propagation speed is completely meaningless**. the wave travelling 1 mesh-interval every 2 iterations, while the Courant number is infinite. The phenomenon is therefore a **numerical pathology**.

Lastly, note that the phenomenon appears only for non-periodic boundary conditions. In the periodic case, all linear operators constant from point-to-point are represented by circulant matrices, and all such matrices can be simultaneously diagonalized by the (same) discrete Fourier transform. This situation is adequate for studying propagation speeds away from boundaries of actual physical waves, or L_2-stability, but inappropriate for iterative convergence rate estimations of a global iteration which should account of boundary conditions.

To illustrate these results, experiments were made with a 1-D finite-volume solver of non-equilibrium chemistry equations of the form

$$\frac{\partial}{\partial t}(\rho_i) + \frac{\partial}{\partial x}(\rho_i V) = \Omega_i \qquad (i = 1,2,3) \tag{40}$$

in which the unknowns are the density functions $\rho_i(x,t)$ (i=1,2,3), V is a constant equal to the normal component of velocity behind a normal shock at a freestream Mach number $M_\infty = 25$, and Ω_i is a nonlinear function of temperature T and ρ_i (i=1,2,3):

$$\Omega_i = \Omega_i(\rho_1, \rho_2, \rho_3, T) \tag{41}$$

Velocity and density functions are related to temperature through the energy conservation equation. The model which accounts for 5 species and 18 reactions will be presented in details in [DES4].

These equations were solved by the linearized Euler implicit method based on a 2nd-order partially-upwind scheme explicitly and a 1st-order upwind scheme implicitly:

$$\left(Id - \Omega'^n \Delta t + \frac{V\Delta t}{\Delta x}\delta^u_{x,1}\right)(W^{n+1} - W^n) = \Delta t\left(\Omega^n - \frac{V}{\Delta x}\delta^\beta_{x,2}W^n\right) \tag{42}$$

in which

$$W^n = \begin{pmatrix} \rho_1^n \\ \rho_2^n \\ \rho_3^n \end{pmatrix} \qquad \Omega^n = \begin{pmatrix} \Omega_1^n \\ \Omega_2^n \\ \Omega_3^n \end{pmatrix} \tag{43}$$

and

$$\Omega'^n = \left(\frac{\partial \Omega}{\partial w}\right)^n \tag{44}$$

is a Jacobian matrix. In the limit $\Delta t \longrightarrow \infty$ the algorithm reduces to:

$$\left(-\frac{\Delta x}{V}\Omega'^n + \delta^u_{x,1}\right)(W^{n+1} - W^n) = \frac{\Delta x}{V}\Omega^n - \delta^\beta_{x,2}W^n \tag{45}$$

Letting now $\Delta x \longrightarrow 0$ yields:

$$\delta^u_{x,1}(W^{n+1} - W^n) = -\delta^\beta_{x,2}W^n \tag{46}$$

Therefore, the preceding theoretical predictions on iterative convergence should closely apply to the algorithm in (43) in the limit $\Delta t \longrightarrow \infty$, $\Delta x \longrightarrow 0$.

To verify this, a mesh of 300 points over 10 cm was constructed, and (43) was applied with the Courant number set equal to 500. (Values of the order of 10^5 could be used stably without noticeable difference in convergence rate.) The convergence history of the norm of the residual (explicit right-hand side) is shown on Figure

2 (a-d) for $\beta = \frac{1}{2}, \frac{1}{3}, 0$ and 1 respectively. Note the different scales for the axis of iterations. These experiments confirm that the former two iterations converge at the rate of the sequence 2^{-n}, achieving the machine-zero in some 40 timesteps, whereas the latter two are pathological, exhibiting a phase of stagnation over some 600 timesteps, that is according to the analysis, twice the number of degrees of freedom.

In conclusion:
When using the slightly inconsistent formulation of the implicit upwind scheme, in which the implicit preconditioner is based on only first-order differencing while a second-order partially-upwind approximation is constructed explicitly, it is not recommended to use the central-differencing scheme ($\beta = 0$) explicitly, or the fully-upwind scheme ($\beta = 1$) explicitly that both result in defective methods with pathological iterative convergence. Preferrably, one should use the half-fully upwind scheme explicitly ($\beta = \frac{1}{2}$) to realize the best separation of the eigenvalues, and presumably the least condition number of the matrix of eigenvectors, or the $\frac{1}{3}$-fully upwind scheme explicitly ($\beta = \frac{1}{3}$) to achieve third-order steady-state accuracy; the latter may however not be as robust.

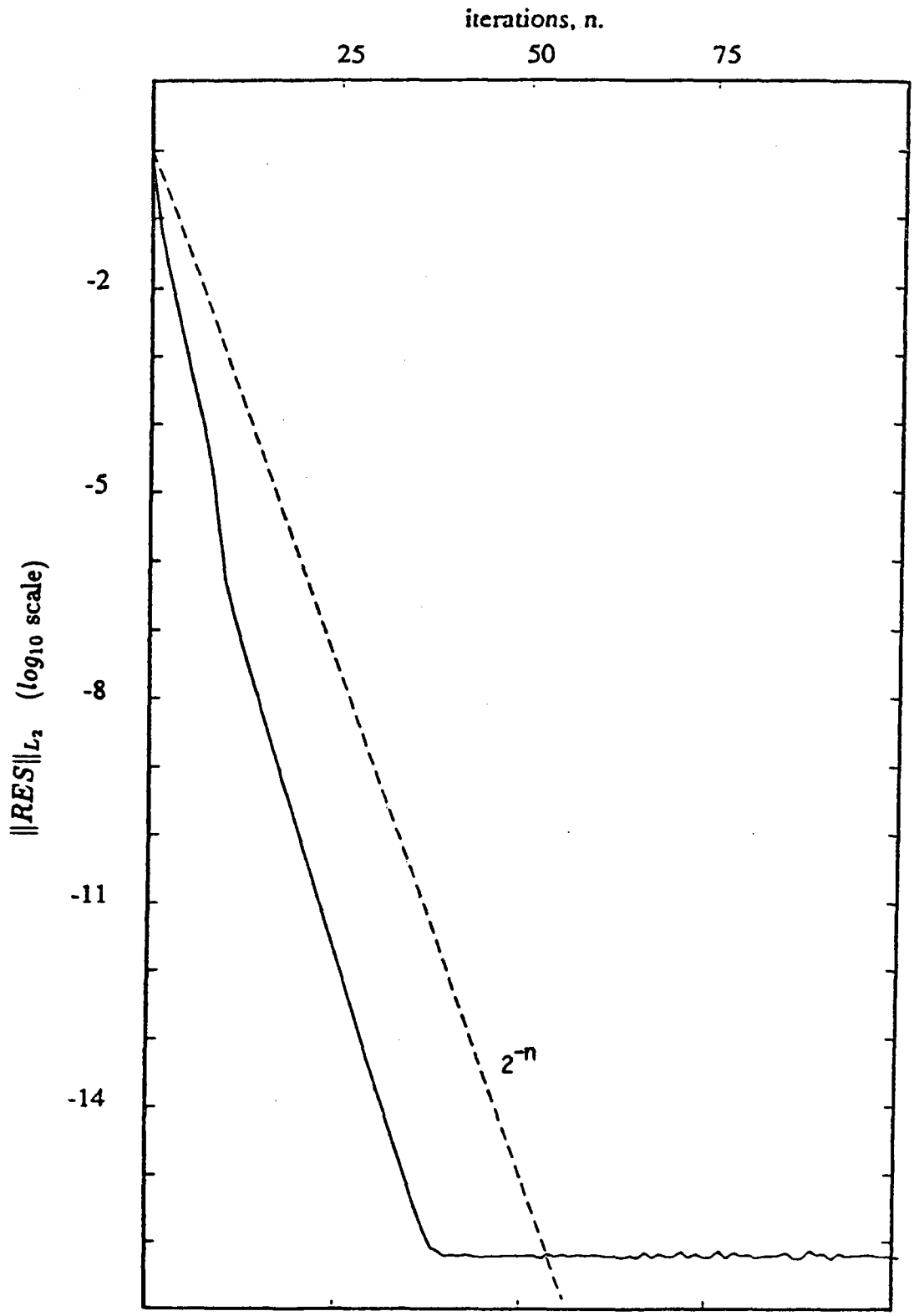


Figure 2 (a) : Convergence history of the 1-D non-equilibrium chemistry implicit solver (CFL= 500)
(a) half-upwind scheme explicitly ($\beta = \frac{1}{2}$)

SCHEMA IMPLICITE DECENTRE D'ORDRE 3 - 1-D (300 pts)

- MACH=25.00 - INCIDENCE= 0.00 -

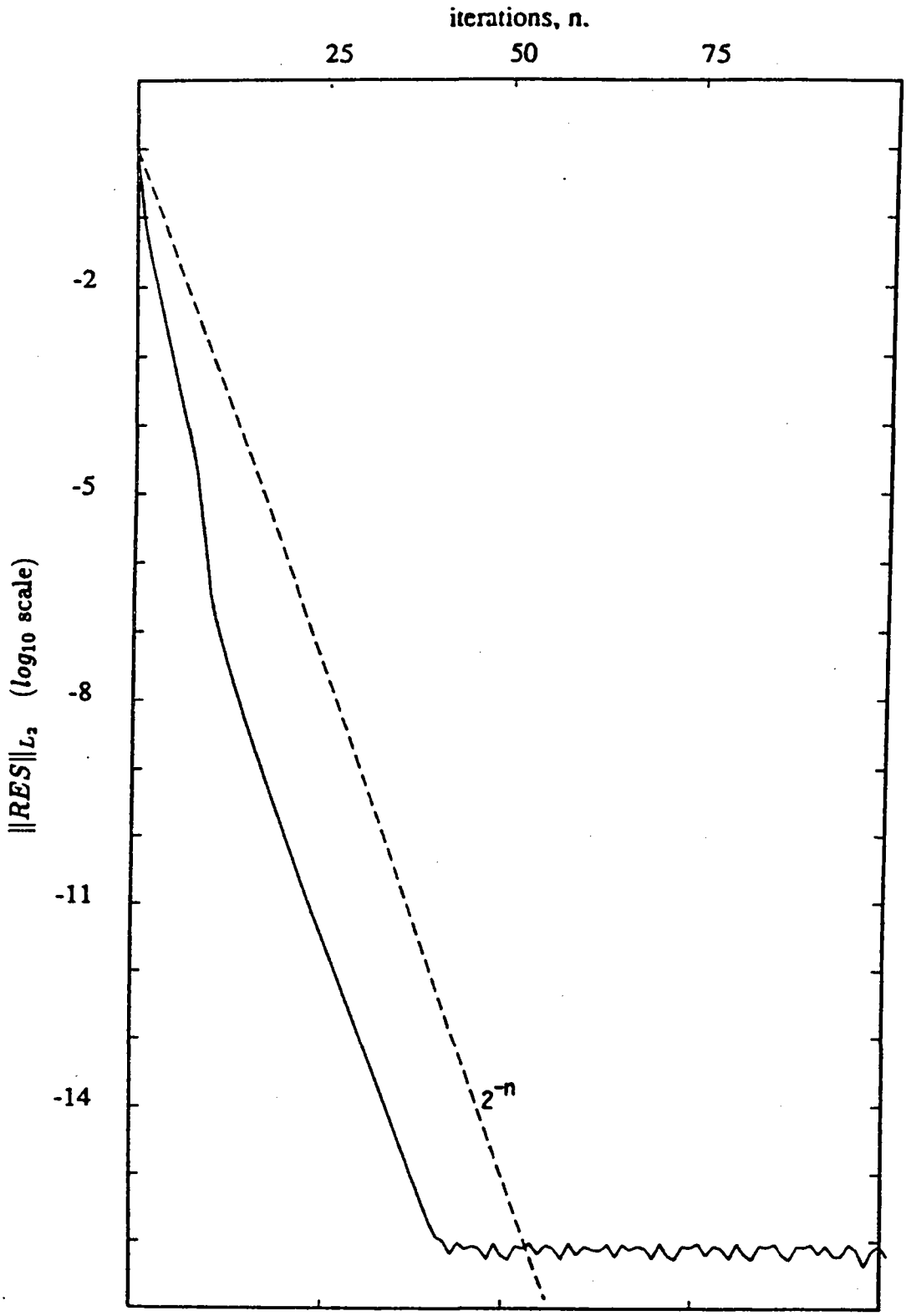


Figure 2 (b) :
(b) $\frac{1}{3}$-fully upwind scheme explicitly ($\beta = \frac{1}{3}$)

SCHEMA IMPLICITE CENTRE D'ORDRE 2 - 1-D (300 pts)

- MACH=25.00 - INCIDENCE= 0.00 -

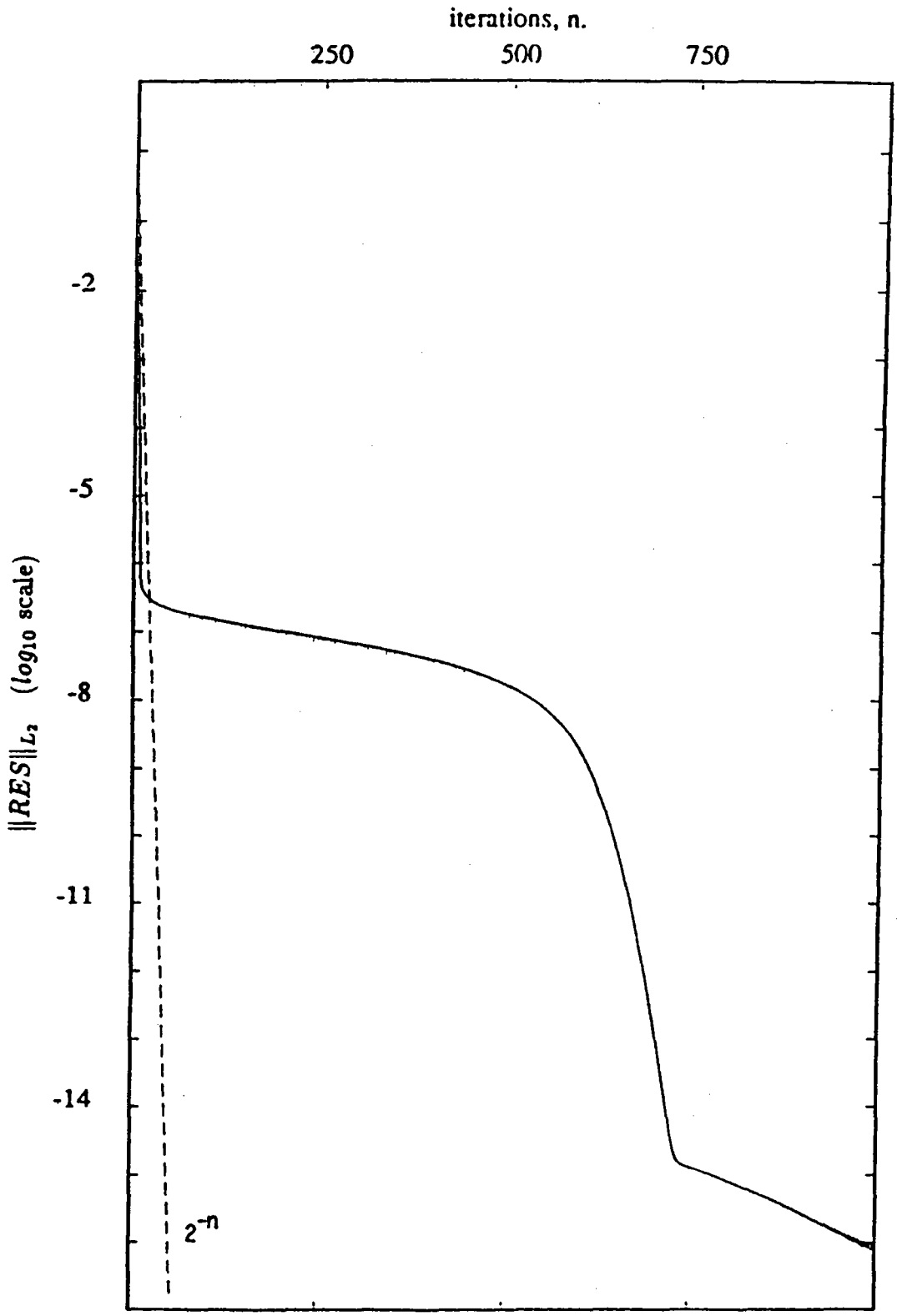


Figure 2 (c) :
(c) central differencing scheme explicitly ($\beta = 0$)

IMPLICITE TOTALEMENT DECENTRE D'ORDRE 2 - 1-D (300p)

- MACH=25.00 - INCIDENCE= 0.00 -

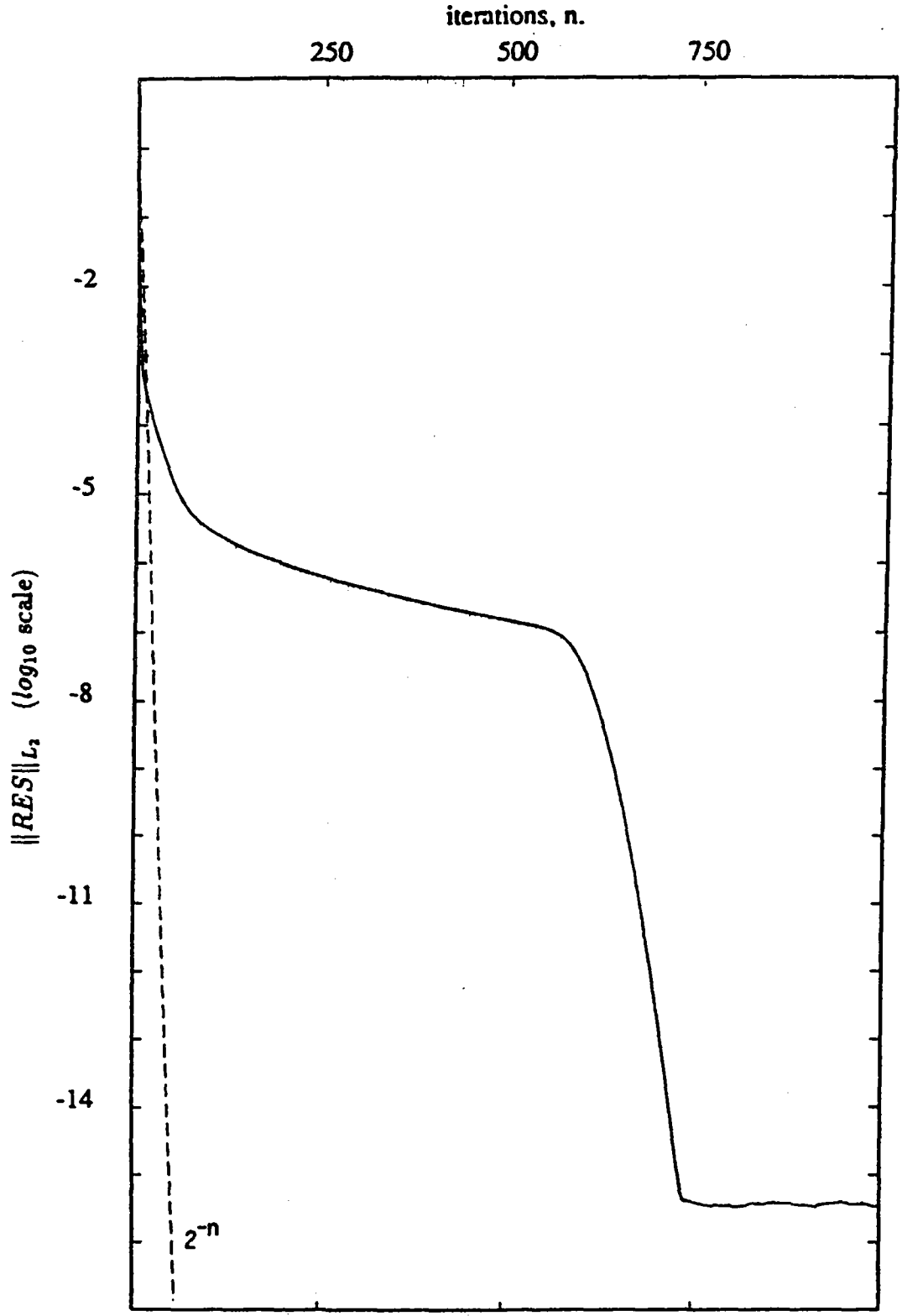


Figure 2 (d) :
(d) fully upwind scheme explicitly ($\beta = 1$)

2. APPLICATION TO THE EULER EQUATIONS

2.1. *The 1-D case*

The Euler equations are written an the following conservation form

$$W_t + (A(W)W)_x = 0 \tag{48}$$

The following H-class of upwind explicit schemes are introduced :

$$\begin{aligned} &\frac{W_i^{n+1} - W_i^n}{\Delta t} + \frac{1}{\Delta x}\left(\Phi^n_{i+\frac{1}{2}} - \Phi^n_{i-\frac{1}{2}}\right) = 0 \\ &\Phi^n_{i+\frac{1}{2}} = \Phi(W_i^n, W_{i+1}^n) \\ &\Phi(U,V) = H_1(U,V)U + H_2(U,V)V \quad for\ U,V \in \mathbb{R}^3 \\ &H_1(U,U) + H_2(U,U) = A(U) \end{aligned} \tag{49}$$

The third line represents a linearization whose precise definition will be given later and the fourth line is a consistency requirement. A "simplified linearized implicit version" of scheme (49) can be defined as follows :

$$\begin{cases} \dfrac{W_i^{n+1} - W_i^n}{\Delta t} + \dfrac{1}{\Delta x}\left(\Phi^S_{i+\frac{1}{2}} - \Phi^S_{i-\frac{1}{2}}\right) = 0 \\ \Phi^S = H_1(U,V)X + H_2(U,V)Z \\ U = W_i^n \ ; \\ V = W_{i+1}^n \ ; \\ X = W_i^{n+1} \ ; \\ Z = W_{i+1}^{n+1} \ ; \end{cases} \tag{50}$$

Scheme (50) can be written in "delta form" :

$$\begin{cases} \sigma\, H^n_{2,i+\frac{1}{2}}\, \delta W_{i+1}^{n+1} + \left[Id + \sigma\, H^n_{1,i+\frac{1}{2}} - \sigma\, H^n_{2,i-\frac{1}{2}} \right] \delta W_i^{n+1} \\ \quad - \sigma\, H^n_{1,i-\frac{1}{2}}\, \delta W_{i-1}^{n+1} = - \sigma \left[\Phi^n_{i+\frac{1}{2}} - \Phi_{i-\frac{1}{2}} \right] \end{cases} \tag{51}$$

with $\sigma = \frac{\Delta t}{\Delta x}$ and where Id denotes the identity operator. This construction results in a scheme producing first-order accurate steady-state-solutions.

Since the H-terms are not exact Jacobians of the approximate fluxes $\Phi(U,V)$, this iterative scheme does not converge quadratically as would Newton's method when $\Delta t \rightarrow \infty$ applied to the discrete system However, very fast convergence can be expected since the method is a consistent discretization of Newton's method applied to the continuous problem.

In fact, for simplicity, the H-terms that we use are derived from the Steger-Warming [STE4] plus-minus flux-splitting

$$H_1(U,V) \;=\; A^+(U) \;\;;\;\; H_2(U,V) \;=\; A^-(V)$$

In this case the H-formulation results in non-differentiable terms .

Spatially second-order accurate scheme : Starting from the delta-formulation (51), an efficient way to obtain second-order accuracy for steady solutions while keeping the interesting properties of the first-order accurate upwind matrix is to replace the R.H.S of (51) by a second-order accurate approximate Euler operator ; if $E_h^2(W)$ denotes such an operator, then the resulting algorithm can be presented as a two-phase scheme :

Phase 1 : Physical / explicit / nonlinear / second-order accurate phase

$$\delta\hat{W}_i \;=\; -\,\sigma\, E_h^2\,(W_h^n) \;\;;\;\; \sigma \;=\; \frac{\Delta t}{\Delta x}$$

Phase 2 : Mathematical / implicit / linear / first-order accurate phase

$$\sigma\, H_{2,i+\frac{1}{2}}\, \delta W_{i+1}^{n+1} \;+\; \left[\, Id \;+\; \sigma\, H_{1,i-\frac{1}{2}}^n \;-\; \sigma\, H_{2,i-\frac{1}{2}} \right]\, \delta W_i^{n+1}$$
$$-\; H_{1,i-\frac{1}{2}}\, \delta W_{i-1}^{n+1} \;=\; \delta\hat{W}_i$$
$$W_i^{n+1} \;=\; W_i^n \;+\; \delta W_i^{n+1} \;.$$

2.2. *The 2-D case*

The complete formulation in the two-dimensional case, including linearization of boundary terms, is written (with the notations of Chapter I):

Phase 1 :

(52)
$$\delta\hat{W}_i \;=\; -\,\frac{\Delta t_i}{Area(C_i)}\; E_h^2(W^n)_i$$

Phase 2 :

$$
\begin{aligned}
&\frac{Area(C_i)}{\Delta t_i}\ \delta W_i^n \\
&+ \left[\sum_{j\ \epsilon\ K(i)} P_{ij}^{+}(W_i^n) \right]\ \delta W_i^n \\
&+ \sum_{j\ \epsilon\ K(i)} P_{ij}^{-}(W_j^n)\ \delta W_j^n \\
&+ \int_{\partial C_i \cap \Gamma_\infty} P_{i\infty}^{+}(W_i^n)\ \delta W_i^n\ d\sigma \\
&+ \int_{\partial C_i \cap \Gamma_B} P_B(W^n)\ \delta W^n\ d\sigma \\
&= \frac{Area(C_i)}{\Delta t_i}\ \delta \hat{W}_i\ , \\
W_i^{n+1} &= W_i^n + \delta W_i^n\ ,
\end{aligned}
\tag{53}
$$

where :

$$
P_B(W) = (\gamma - 1)
\begin{pmatrix}
0 & & & \\
\frac{1}{2}(u^2+v^2+W^2)\nu_x & 0 & 0 & 0 \\
\frac{1}{2}(u^2+v^2+W^2)\nu_y & -u\nu_x & -v\nu_x & \nu_x \\
0 & -u\nu_y & -v\nu_y & \nu_y \\
 & 0 & 0 & 0
\end{pmatrix}
\tag{54}
$$

is derived from the linearization of the body boundary condition ; Δt_i holds for a local time-step, providing faster convergence to steady-state.

Fastly converging time iteration

As noticed by van Leer and Mulder [LEE3], fast convergence can be obtained by applying large time-steps after a few steps of a "solution search " phase.

In fact the best efficiency will be the result of adhoc trade-offs in tuning the time-step and the termination criterion of the iterative solution of the linear system (the system need not be brought to full convergence).

A typical example is the calculation of a transonic flow around a NACA0012 airfoil (Mach at infinity is .85, no angle of attack): large time-steps cannot be used

stably during the first few time steps ; however, when the number of Gauss-Seidel sweeps is maintained equal to 20, the time-step can be made larger ; the following empirical rule is applicable :

$$CFL = Max(1, \frac{10}{RES})$$

where the "RES" is a quadratic norm of the E_h^2 term in (52). In these conditions, convergence is obtained in 10-20 time steps , with an asymptotic reduction factor in agreement with the conjecture of Section 1 (see also Chapter III).

Application to vector computing

The application of Gauss-Seidel iteration does not permit vector computing on usual pipeline computers neither on parallel ones because of the data dependencies involved in this method: in each sweep, each node calculation requires the result of several neighboring ones.

One way to circonvent this difficulty is to employ the Jacobi iteration instead. Then convergence is slowed by a factor of about 2 in the usual transonic cases, but the global efficiency of the method is improved due to vectorization. In practice, at least 4 Jacobi sweeps are necessary to maintain stability with large CFL numbers.

However, It should be stressed that the performance of the Jacobi version is severely degraded when it is applied to low Mach calculations [BEN].

Further improvement w.r.t. the Jacobi iteration may be expected when a more sophisticated iterative scheme such as Red-Black Gauss-Seidel is applied however we have no experience of it (note that, when using finite-element meshes, the number of colors should be more than 2).

Low-Storage vector version

The use of the first generation of supercomputers put in evidence an important memory storage bottle-neck in the application of unfactored implicit methods in which full multi-dimensional matrices are stored. For the second generation (CRAY-2), a massive access to the memory can result in lower performances.

One way to deal with this problem is to use a low-storage linear iteration which consisting of re-computing off-diagonal terms at each node each time they are required.

Actually, only "right-hand sides" are needed and their calculation is essentially equivalent to assembling an explicit (first-order) approximation.

Then the number of operations is increased and the efficiency is degraded, but the storage is restricted to the inverses of the (block-) diagonal elements.

Note that in this low-storage context, a red-black iteration would be about twice as costly as the Jacobi iteration since , for each segment (joining two nodes), the flux would be computed twice; now the red-black is not likely to converge twice faster than the Jacobi iteration so that the red-black algorithm a priori seems less attractive in this context.
A comparison of the different efficiencies for a typical case is presented in Table 1.

3. A SAMPLE OF 3-D EXPERIMENTS

Euler (pseudo-)time marching calculations permit the prediction of a large range of flow regimes and angles of attack, possibly involving separation and recirculation zones. Starting from a structured "Finite Difference-like" surface mesh on the body, a non-structured tetrahedrization is generated by a quasi-automatic process. This approach uses a smaller number of nodes than usual structured mesh generation . For each geometry, the mesh contains about 10,000 nodes (half-geometry, zero yaw angle). Since the number of points is not too large, many cases can be calculated in a short delay with a rather good solution accuracy.

The first geometry corresponds to the U.S. Orbiter ; the second one is a shape designed for the European Hermes shuttle project (Fig. 3-8).

For each case the implicit calculation yields a converged solution within less than 200 time-steps and requires approximately 3 hours of CPU on a IBM 3090.

The accuracy level of the results obtained is sufficient to reach a good agreement of global aerodynamical coefficients (lift,pitching moment) with available experimental datas.

4. NOTES

The first section is taken from [DES2] ; see also [DES1]. The full storage scheme was derived by Fezoui and Stoufflet ([FEZ3]). The low storage algorithm was derived by H. Steve [STE] ; another low storage scheme is proposed in [EBE] ; the alternative is dissussed in [STE]. The 3-D calculations were presented in [STO3]. The CFL law is close to the one introduced in the Switch Evolution Relaxation scheme of van Leer and Mulder [LEE3].

For vector computing, we choose the explicit-like point-jacobi iteration ; this will also permit parallelisation ; in the structured case, many other strategies are possibles, we refer to the other Notes of this series.

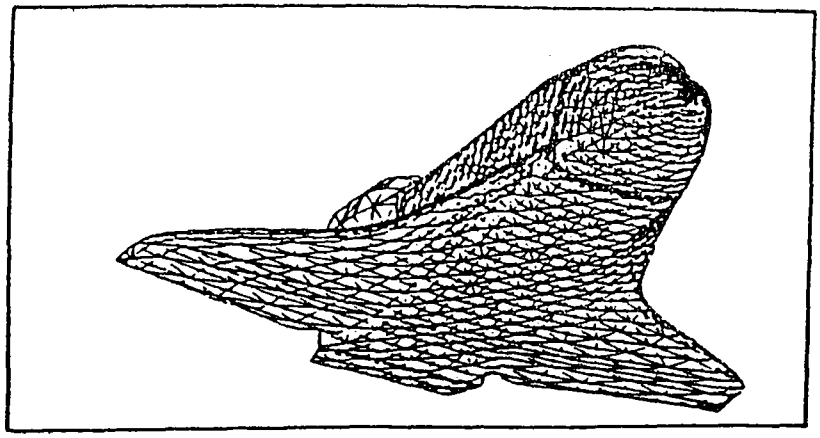
TABLE 1 : EFFICIENCY COMPARISON

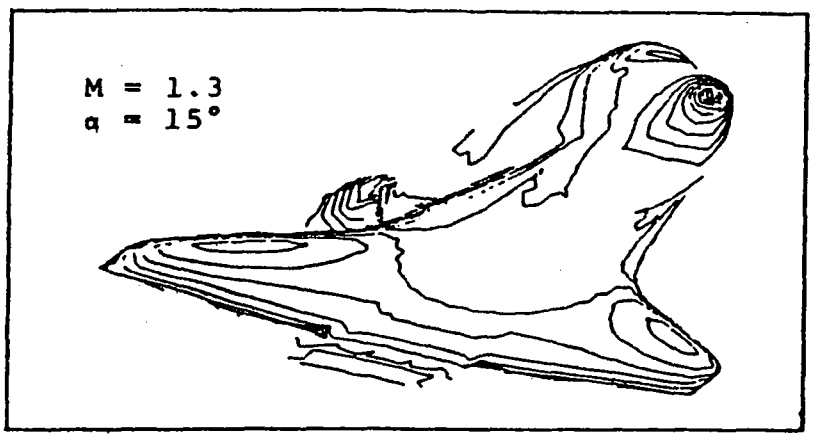
BETWEEN THREE SOLVERS

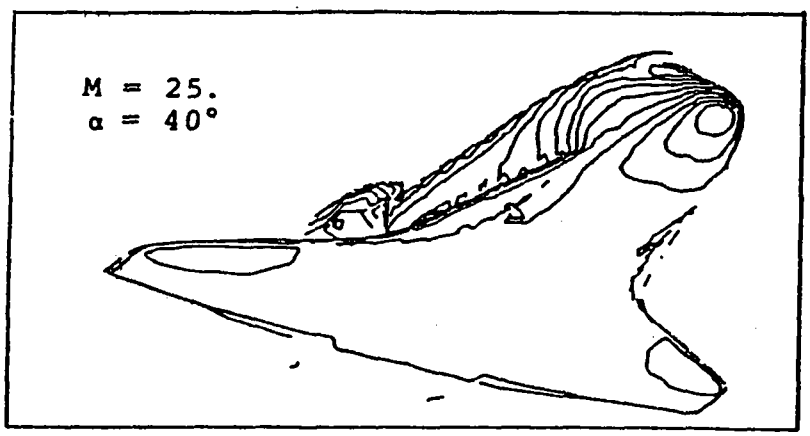
- EXPLICIT
- IMPLICIT FULL STORAGE (WITHOUT MATRIX FREEZING)
- IMPLICIT LOW STORAGE

SCHEME	CPU FOR 1 ITERATION Order	(*)	CPU FOR SOLUTION (**)	RATIO	STORAGE (***)
EXPLICIT	1	12 sec	318 min	1	100 x NS
CFL = 0,6	2	16 sec	373 min	1	
IMPLICIT GAUSS-SEIDEL FULL STORAGE	1	54 sec	57 min	4	212 x NS
	2	71 sec	106 min	3,5	
IMPLICIT JACOBI LOW STORAGE	1	78 sec	76 min	3	116 x NS
	2	104 sec	156 min	2,5	

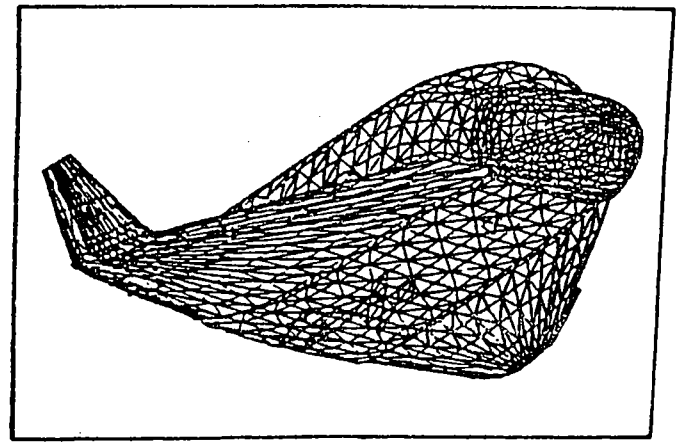
(*) seconds DPS-68
(**) Residual divided by 10000, minutes DPS-68
(***) NS = number of vertices

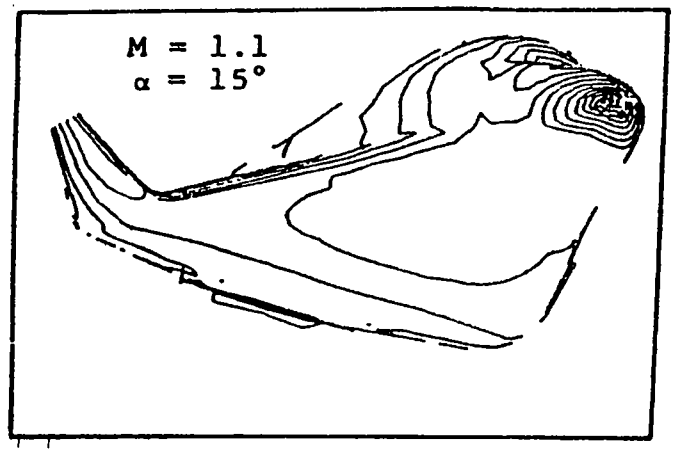


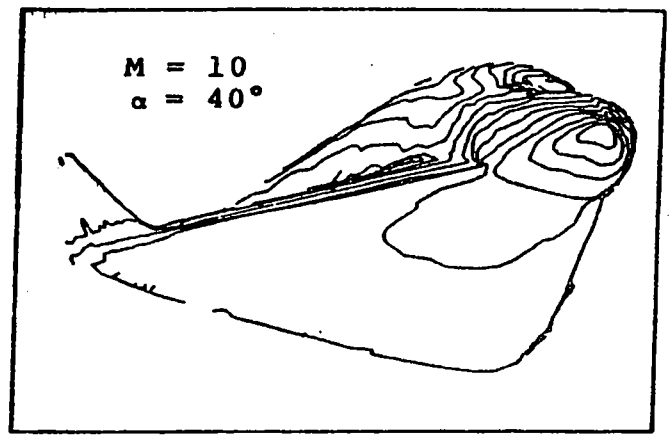




FIGURES 3 to 5







FIGURES 6 to 8

5. APPENDIX I : MODEL DIFFERENCING SCHEMES

Model Differencing Schemes

matrices are $N \times N$ ($N = 25$ for plots)
$\theta_m = \frac{2m_i\pi}{N}$, $m = 0, 1, \cdots, N-1$
LB/RB: left/right boundary procedure

(1) central-difference operator

(a) periodic case (o)

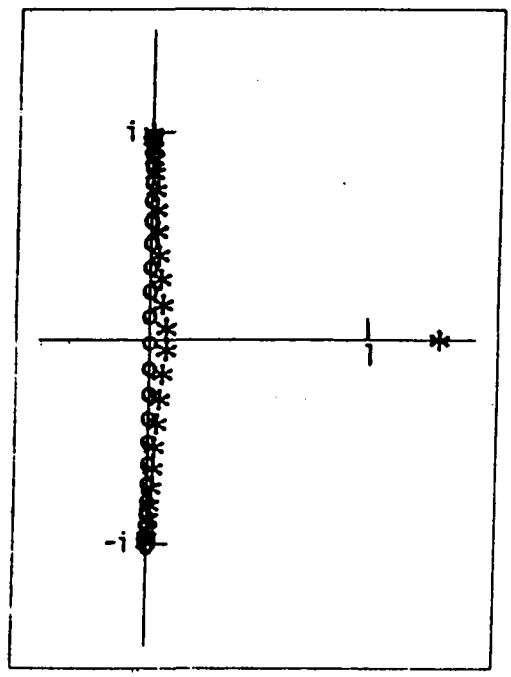
$$\delta_x^c = \frac{1}{2}\begin{pmatrix} 0 & 1 & & & & -1 \\ -1 & 0 & 1 & & & \\ & -1 & 0 & 1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & -1 & 0 & 1 \\ 1 & & & & -1 & 0 \end{pmatrix}$$

skew-symmetric (diagonalizable) matrix
purely imaginary eigenvalues
$\lambda_m = i \sin\theta_m$
$Re(\lambda_m) = 0$

(b) non periodic case (*)
LB: Dirichlet (u_0 given): RB: 1st-order upwind

$$\delta_x^c = \frac{1}{2}\begin{pmatrix} 0 & 1 & & & & \\ -1 & 0 & 1 & & & \\ & -1 & 0 & 1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & -1 & 0 & 1 \\ & & & & -2 & 2 \end{pmatrix}$$

(diagonalizable matrix)
$\lambda_0 \in \mathbb{R}^+$
and for $m = 1, 2, \cdots, N-1$
$Re(\lambda_m) > 0$. and ≈ 0



(2) 1st-order upwind (backward-difference operator)

(a) periodic case (o)

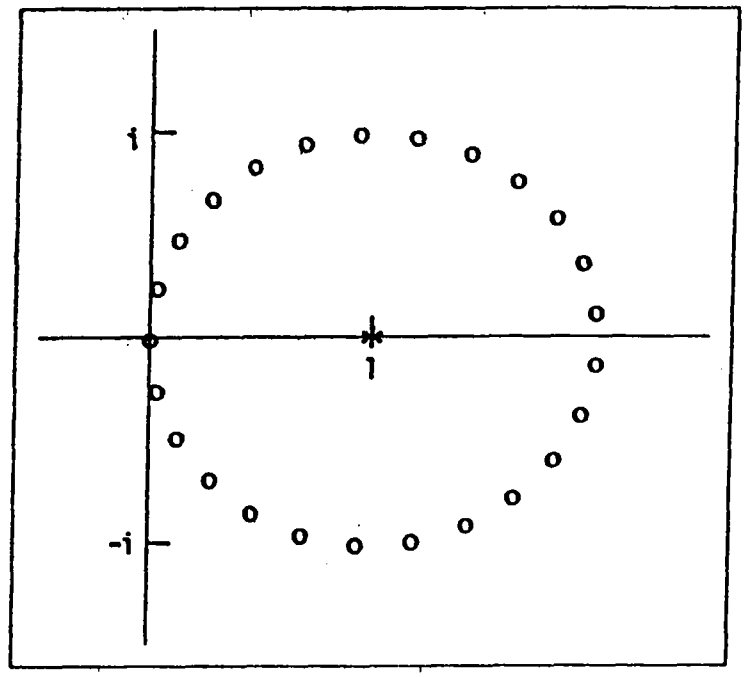
$$\delta^{u}_{x,1} = \begin{pmatrix} 1 & & & & & -1 \\ -1 & 1 & & & & \\ & -1 & 1 & & & \\ & & \ddots & \ddots & & \\ & & & -1 & 1 & \\ & & & & -1 & 1 \end{pmatrix}$$

(diagonalizable matrix)
$\lambda_m = 1 - e^{-i\theta_m}$

(b) non periodic case (*)
LB: Dirichlet (u_0 given)

$$\delta^{u}_{x,1} = \begin{pmatrix} 1 & & & & & \\ -1 & 1 & & & & \\ & -1 & 1 & & & \\ & & \ddots & \ddots & & \\ & & & -1 & 1 & \\ & & & & -1 & 1 \end{pmatrix}$$

(defective) triangular matrix
$\lambda_m = 1, \forall m$



(3) 2nd-order fully upwind (backward-difference operator)

(a) periodic case (o)

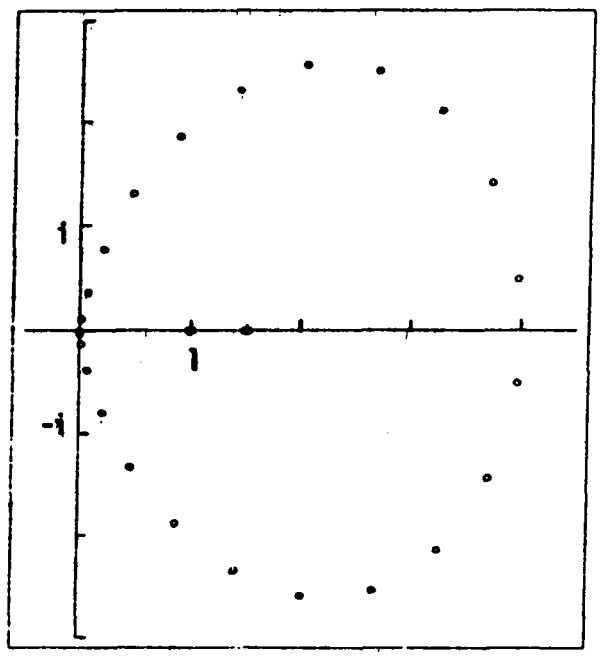
$$\delta^u_{x,2} = \frac{1}{2}\begin{pmatrix} 3 & & & & 1 & -4 \\ -4 & 3 & & & & 1 \\ 1 & -4 & 3 & & & \\ & \ddots & \ddots & \ddots & & \\ & & 1 & -4 & 3 & \\ & & & 1 & -4 & 3 \end{pmatrix}$$

(diagonalizable matrix)
$\lambda_m = \frac{1}{2}(3 - 4e^{-i\theta_m} + e^{-2i\theta_m})$

(b) non periodic case (*)
LB: Dirichlet (u_0 given), and 1st-order upwind scheme applied at gridpoint 1

$$\delta^u_{x,2} = \frac{1}{2}\begin{pmatrix} 2 & & & & & \\ -4 & 3 & & & & \\ 1 & -4 & 3 & & & \\ & \ddots & \ddots & \ddots & & \\ & & 1 & -4 & 3 & \\ & & & 1 & -4 & 3 \end{pmatrix}$$

(defective) triangular matrix
$\lambda_0 = 1, \lambda_m = \frac{3}{2}$ $(m = 1, 2, \cdots, N-1)$



6. APPENDIX II : DEFECTIVE LINEAR ITERATIONS

In this appendix, we examine the case of a general linear non-homogeneous iteration,

$$W^{n+1} = GW^n + b \tag{55}$$

in which the unknown w is an N-vector, G a given $N \times N$ amplification matrix and b is a given constant N-vector. in the particular case where **the matrix G is defective, that is, cannot be diagonalized.**

By a suitable similarity transform, G can still be reduced to the so-called **"Jordan canonical form"** [STR]:

$$J = X^{-1}GX \tag{56}$$

where X is the generalized eigenvector matrix, and J is block-diagonal:

$$J = BDiag(J_i) = \begin{pmatrix} J_1 & & & \\ & J_2 & & \\ & & \ddots & \\ & & & J_s \end{pmatrix} \tag{57}$$

where s is the number of linearly independent (true) eigenvectors ($s < N$), and each block J_i has the following bidiagonal structure:

$$J_i = Bidiag(1, \lambda_i) = \begin{pmatrix} \lambda_i & & & & \\ 1 & \lambda_i & & & \\ & 1 & \lambda_i & & \\ & & \ddots & \ddots & \\ & & & 1 & \lambda_i \end{pmatrix} \tag{58}$$

The matrix J being triangular, it contains its eigenvalues in its main diagonal; these are the numbers λ_i ($i = 1, 2, .., s$) that also are the eigenvalues of the matrix G to which J is similar. To assure convergence of the iteration, it is assumed that the amplification matrix G satisfies the spectral radius condition [VAR]:

$$\rho(G) = \max_{i=1,2,\cdots,s} (|\lambda_i|) \tag{59}$$

In such case, none of the eigenvalues of the matrix G is equal to 1, the matrix $I - G$ is invertible, and the iteration admits a fixed-point solution $W^\infty = (I - G)^{-1}b$ that satisfies

$$W^\infty = GW^\infty + b \tag{60}$$

Then defining the "error vector" e^n by

$$e^n = W^n - W^\infty \tag{61}$$

and subtracting (55) from (60) it follows that e^n satisfies the following homogeneous linear iteration

$$e^{n+1} = Ge^n \tag{62}$$

which implies that

$$e^n = G^n e^0 \tag{63}$$

But,

$$G^n = XJ^nX^{-1} \tag{64}$$

and in the basis of the generalized eigenvectors, the error vector becomes,

$$\epsilon^n = X^{-1}e^n \tag{65}$$

so that combining (63)-(65) yields the expression:

$$\epsilon^n = J^n\epsilon^0 \tag{66}$$

In addition, as a consequence of the block-diagonal structure of the matrix J given by (57),

$$J^n = BDiag(J_i^n) \tag{67}$$

It is therefore apparent that the attenuation with increasing n of the error-vector components is governed by the powers of the individual blocks J_i taken separately. For this reason, in what follows, and without great loss of generality, we consider the case of only one block ($s = 1$; $J = J_1$; $\lambda_1 = \lambda$). Using (58) several times successively, we obtain:

$$J^2 = \begin{pmatrix} \lambda^2 & & & & \\ 2\lambda & \lambda^2 & & & \\ 1 & 2\lambda & \lambda^2 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & 2\lambda & \lambda^2 \end{pmatrix} \tag{68}$$

(69)
$$J^3 = \begin{pmatrix} \lambda^3 & & & & & \\ 3\lambda^2 & \lambda^3 & & & & \\ 3\lambda & 3\lambda^2 & \lambda^3 & & & \\ 1 & 3\lambda & 3\lambda^2 & \lambda^3 & & \\ & \ddots & \ddots & \ddots & \ddots & \\ & & 1 & 3\lambda & 3\lambda^2 & \lambda^3 \end{pmatrix}$$

More generally, and for $n < N$, the block J^n is a banded lower-triangular matrix having n nonzero subdiagonals below the main diagonal:

(70)
$$J^n = \begin{pmatrix} \lambda^n & & & & & & & & & \\ n\lambda^{n-1} & \lambda^n & & & & & & & & \\ C_n^2\lambda^{n-2} & n\lambda^{n-1} & \lambda^n & & & & & & & \\ \vdots & \ddots & \ddots & \ddots & & & & & & \\ C_n^j\lambda^{n-j} & \cdots & C_n^2\lambda^{n-2} & n\lambda^{n-1} & \lambda^n & & & & & \\ \vdots & \ddots & & \ddots & \ddots & \ddots & & & & \\ 1 & \cdots & C_n^j\lambda^{n-j} & \cdots & C_n^2\lambda^{n-2} & n\lambda^{n-1} & \lambda^n & & & \\ 0 & \ddots & & \ddots & & \ddots & \ddots & \ddots & & \\ \vdots & \ddots & \ddots & & \ddots & & \ddots & \ddots & \ddots & \\ 0 & \cdots & 0 & 1 & \cdots & C_n^j\lambda^{n-j} & \cdots & C_n^2\lambda^{n-2} & n\lambda^{n-1} & \lambda^n \end{pmatrix}$$

where $C_n^j = \frac{n!}{j!(n-j)!}$. Evidently, each nonzero entry of the matrix J^n is one of the monomials appearing in the expansion of $(1+\lambda)^n$.

The first column vector of the matrix J^n is precisely the value achieved by ϵ^n if ϵ^0 is set to $e^1 = (1, 0, 0, \cdots, 0)^T$, that is the first vector of the canonical basis. Since one of its components (the $n+1$st) is equal to 1, its sup-norm (or maximum component in absolute value) is greater or equal to 1, and this, over at least $N-1$ iterations. Therefore, in the first $N-1$ iterations the process cannot be expected to be dissipative, even in the particular case where the spectral radius is equal to 0 ($\lambda = 0$).

For $n \geq N$, the above vector is truncated to the first N components. In particular, its last component becomes $C_n^{N-1}\lambda^{n-N+1}$. This term, for fixed N, is equal to the value at n of a monomial of degree $N-1$ times λ^n. The other components involve monomials of lesser degrees. Therefore, in the most general case, the vector ϵ^n is a general linear combination of the column-vectors of the matrix J^n and any of its components, say ϵ_j^n, is of the form $P_j^{N-1}(n)\lambda^n$ where P_j^{N-1} is a polynomial of degree $N-1$ depending on j. Consequently, as $n \longrightarrow \infty$, $\|\epsilon^n\|_\infty \longrightarrow 0$ only at the rate of $n^{N-1}\rho^n$. From this we conclude that relatively to the regular case for which the amplification matrix G can be diagonalized and the error tends to 0 at the same rate as ρ^n, the asymptotic convergence is slightly degraded by the additional factor

n^{N-1}. Unfortunately, another form of degradation of the convergence, believed to be more severe than the first, is now going to be demonstrated. For this, let:

$$\tau_n = \|\epsilon^n\|_\infty = \max_{j=0,1,\cdots,N-1} (\xi_j^n) \tag{71}$$

in which

$$\xi_j^n = C_n^j \rho^{n-j} \tag{72}$$

in which again, $\rho = \rho(G) = |\lambda| < 1$ is the spectral radius. It turns out that the asymptotic convergence rate is significant only after a relatively large number of iterations. To illustrate the phenomenon, we begin with a numerical experiment. The sequence τ_n was evaluated in the particular case where $\rho = \frac{1}{2}$, $N = 20$ and $\epsilon^0 = (1, 0, \cdots, 0)^T$. The result of this calculation is shown on Figure 9 on a semi-logarithmic plot, where the sequence 2^{-n} is also represented. There it appears that the sequence τ_n is firstly constant and equal to one over a few iterations, then it is monotone-increasing over a number of iterations equal to about 40, that is $2N$, and finally it decreases, producing a parabolic branch admitting no asymptote but instead, an asymptotic direction given by the straight line representing 2^{-n} on this plot. (This agrees with the asymptotic convergence rate previously determined.) Also note that when τ_n achieves its maximum, the value of ρ^n is yet close to about 10^{-10} indicating that the spectral radius in this experiment was chosen relatively small, a generally very favorable circumstance. To explain these observations, we return to the general case and verify that:

- **For N large, it takes the sequence τ_n a number of iterations equivalent to $N/(1-\rho)$ to become monotone-decreasing.**

In this analysis, N is large and since $n > N$, n is also large, a fortiori. First examine the variation with j of ξ_j^n for fixed n. We have:

$$\frac{\xi_j^n}{\xi_{j-1}^n} = \frac{n-j+1}{j\rho} > 1, \quad \text{iff} \quad j < \frac{n+1}{\rho+1} \tag{73}$$

Therefore, for fixed n, the sequence ξ_j^n $(j = 0, 1, \cdots)$ is initially monotone increasing, and only eventually decreases if values of j greater than $\frac{n+1}{\rho+1}$ exist (knowing that $j \leq N-1$). This leads us to analyze two cases separately:

1st case: $N < n < (\rho+1)N - 1$ (N large)
In view of (73), it appears that since the ratio $(n+1)/(\rho+1)$ is less than N, it is

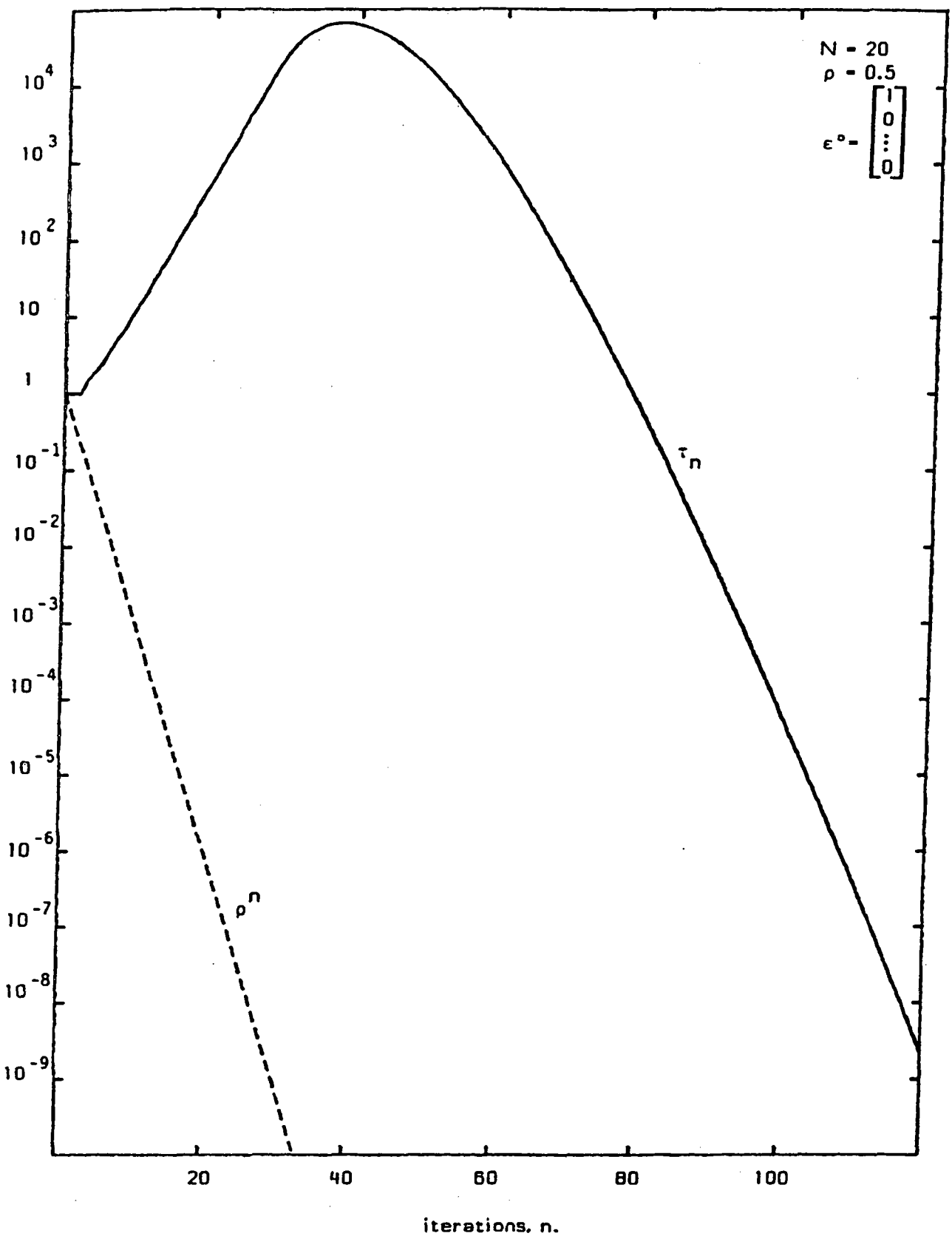


Figure 9 :Variation of τ_n with n compared with ρ^n

for a value j_0 of j close to that ratio that ξ_j^n achieves its maximum over j, which implies that:

$$\tau_n = \xi_{j_0}^n$$

The statement in (73) also indicates that if n increases of 1, the maximum will be achieved at $j_1 = j_0$, or possibly $j_0 + 1$. Therefore:

$$\tau_{n+1} = \xi_{j_0}^{n+1}, \quad \text{or possibly} \quad \xi_{j_0+1}^{n+1}$$

In both cases, using $j_0/n \approx 1/(\rho + 1)$, one obtains that

$$\frac{\tau_{n+1}}{\tau_n} \approx \rho + 1 > 1$$

and therefore n is found insufficiently large for the sequence τ_ν $(\nu \geq n)$ to be monotone-decreasing.

2nd case: $n \geq (\rho + 1)N - 1$ (N large)
Then (73) indicates that the sequence ξ_j^n monotonically increases with j. Therefore, for all $\nu \geq n$,

$$\tau_\nu = \xi_{N-1}^\nu = C_\nu^{N-1} \rho^{\nu - N + 1}$$

and

$$\frac{\tau_{\nu+1}}{\tau_\nu} = \frac{(\nu + 1)\rho}{(\nu + 1 - N + 1)} = \frac{\rho}{1 - \frac{N-1}{\nu+1}}$$

and this ratio is less than 1 for all $\nu \geq n \iff$

$$n > \frac{N - 2 + \rho}{1 - \rho} \sim \frac{N}{1 - \rho}$$

□

In summary:
If a linear iteration has a defective amplification matrix G, still satisfying the spectral radius condition, $\rho < 1$ that insures convergence, then for a general initial guess, the asymptotic convergence will only be like $n^{N-1}\rho^n$, where N is the dimension of the larger Jordan block appearing in the reduction of the amplification matrix. Moreover, if the number N is large, indicating that a large number of eigenvectors are missing, the asymptotic convergence rate is meaningful only after a number of iterations of the order of $N/(1 - \rho)$, a particularly severe degradation if N is very large or if ρ is close to unity, or both.

7. APPENDIX III : ADAPTATION TO VECTOR COMPUTING, SOME BASIC REMARKS

The adaptation to vector computing of the above method is done in three phases

(1) Basic choices in approximation This phase only consists in choosing an homogeneous enough approximation ; we sum it with two notions :
- Taking low-order Finite Element degrees of freedom in order to minimize the number of different approximation molecules ; the P1, vertex formulation is very close to finite volume.
- Considering conformal meshes : even when mesh refinement is performed (see Chapter IV), the mesh remains conformal (as defined in Chapter I) so that the data structure saves some regularity.

(2) Basic choices in solution algorithm When an implicit algorithm is used, then most part of the computational effort lies in the linear system solution. Three criteria in the designing of the linear solver can be considered
(i) dependency in the iteration sweeps result in scalar speed
(ii) scatter and gather are not fast vector operations
(iii) contignous vector operations are fast.

We resrict our discussion to iterative methods * : we derive from (i) that Jacobi or Red-Black Gauss Seidel one ;
From (ii), we have to minimize the scatter gather operations (this favour implicit algorithms)
Startement (iii) favours block-diagonal precondition, as in Jacobi iteration.

Furthermore, the application of the Multi-grid technique will provide a faster convergence without loosing vector or parallel abilities.

(3) Adapted coding In a Compressible CFD vector code, every interior loop must be a long one, and must be vectorizing, except in some steps of a short preparating phase, (one can imagine that parts the preparation phase are performed one for several calculations). In the unstructured case, most of interior loop are as long as the number of nodes (or elements ...)

In [ANG5], F. Angrand and J. Erhel give the following recommendations :
- Handle scatter and gather separately
- Force vectorization of scatter and gather after construction of coloured numbering.

---

* Direct methods can become competitive in the future

CHAPTER III : A MULTIGRID FINITE ELEMENT METHOD

This chapter is essentially devoted to the "unstructured aspects" of multigridding on finite element meshes. Therefore two important points in MG-FEM will not be discussed :
- Grid transfers in finite element between embedded grids : we refer for example to [HIR1], [PER1].
- The adaptation of the iterative method to upwind MUSCL-type schemes, we refer to [TUR], [LAL2].

Instead, we shall focus on the method used to generate the different grids : two families of methods can be considered :

Topological methods

Generally they are based on refinement. Starting from an unstructured coarse grid, finer grids are generated by element division either over the whole computational domain or locally only, after a posteriori error estimates. We refer to [ANG6], [BAN], [PER1], [PER2] [ROS], [LEC] for studies using these two points of view. These topological methods cannot handle the case of an a priori given fine grid ; then the quality of the fine grid is too much dependent on the initial coarse grid.

Another topological approach consists in generating separately non embedded grids ([LOH1], [MAV]).

Algebraic methods

Starting from a linear system derived from an arbitrary unstructured fine mesh formulation, coarser levels are generated by gathering related equations (lines of the matrix) ; we refer to [BRA].

Algebraic Multigrid Methods are usually applied to linear systems and necessitate the construction and generally also the storage of the matrix.

We propose a new topological approach with the following features :

- the grids are embedded ;

- the coarse meshes are not classical FEM triangulations but generalized finite volume partitions ;

- the spatial approximation is derived on each level ;

- a full approximation storage (non linear) scheme is employed.

1. THE GENERATION OF THE DIFFERENT LEVELS

The objective is to generate coarse levels automatically from an arbitrary unstructured triangulation.

To achieve such degree of reliability we explore the possibility of grouping together nodes associated with contiguous control volumes. Thus, coarse levels are not produced by a new **triangulation** of the domain. However, identifying nodes to **control volumes** permits a homogeneous description of the different levels in terms of Finite Volume partitions.

Finite Volume Dual mesh :

Indeed, it has been observed that simplicial (triangles,tetrahedra) Galerkin approximations are equivalent in some sense to adhoc finite volume formulations on specific dual meshes : for the two-dimensional case, the dual mesh is derived using the medians of the triangles.

Coarsening agglomerating algorithm :

Grouping together control volumes results in a new (coarser) mesh. Repeating the operation allows us to get coarser and coarser levels until sufficiently many levels are obtained.

An algorithm for grouping cells together should satisfy the following criteria :

(1) The size of the mesh should decrease while the maximum allowable time step (for explicit iterations) should increase.

(2) The solution should be sufficiently accurately represented on coarser grids to obtain a good initialization (Full-MG) and good preconditioners ,

(3) The sequence of nested grids should allow the damping of a dense enough collection of frequency modes,

(4) The procedure should not be costly.

One approach could consist of using some auxiliary regular coarser mesh which divides the domain in regions, in order to gather the cells whose centers belong to the same region. Such an approach may not sufficiently account for the density of the initial mesh .

Some more sophisticated methods could be considered : we could derive them from the works motivated by multi-tasking in super-computers ; the problem is to divide the domain in regions which are (1) of comparable size (number of nodes) and (2) with as few connections between each other as possible (therefore with as much connection in each region). The sophistication can be increased up to the examination of the discrete equation as in Algebraic MG methods.

The coarsening algorithm that we present here is based on neighboring relations (two cells are neighbors if they contain vertices that are neighbors) ; the algorithm reads as follows :

Consider successively every cell.
(1) if the cell C is already included in a group then consider next cell; else: create a new group containing C, and put into this group each cell neighboring C and not already included in a group.
(2) if the new group contains only the cell C, then destroy the group and put cell C in an existent group containing a neighbor of C.
(3) next cell.

Note that one obvious disadvantage of this method is that coarse levels can destroy the possible symmetry properties of the fine mesh.

2. GENERALIZED FINITE VOLUME UPWIND SCHEMES

One main feature of the algorithm is that it relies on the construction of a Finite-Volume Method applicable to an arbitrary partition of the computational domain. This construction is presented for a first-order accurate upwind scheme.

2.1. *First-order scheme*

The time-dependent Euler equations are written in conservative form :

$$W_t + F(W)_x + G(W)_y = 0.$$

in which as usual :

$$W = (\rho, \rho u, \rho v, E)$$

where ρ is the density, (u, v) the velocity and E the total energy per unit volume.

The upwind finite volume scheme is derived in the simplest manner that one can imagine. We describe it in the context of the usual explicit time-stepping.

Given a cell C_i , the mean value W_i of the dependent variable in this cell is advanced from time level n to time level $n+1$ as follows :

$$area(C_i)[W_i^{n+1} - W_i^n] = -\Delta t \sum_{j \ neighbor \ of \ i} \Phi(W_i^n, W_j^n, \eta^{ij})$$

where $\vec{\eta}^{ij}$ is the following metric vector :

$$\eta_x^{ij} = \int_{\partial C_i \cap \partial C_j} \nu_x d\sigma$$

$$\eta_y^{ij} = \int_{\partial C_i \cap \partial C_j} \nu_y d\sigma$$

$\vec{\nu} = (\nu_x, \nu_y)$unit normal vector pointing outward from C_i

and where Φ is a flux-splitting consistent with $\eta_x^{ij} F + \eta_y^{ij} G$, as explained in Chapter I.

On the finest mesh, that is a standard triangulation, either the first order scheme, or the second order scheme defined in Chapter 1 is applied.

2.2. *Stability*

The efficiency, and to some extent the robustness of the algorithm relies on the accurate estimation of the maximal time-step ; this is particularly essential when local time-stepping is employed.

Unfortunately, estimating the local time-step evaluated from a simplified Fourier analysis can be very hazardous. Hence, we prefer to evaluate a lower bound based on the L^∞ stability of a two-dimensional model.

Then two models can be useful to the study :

(1) - Constant-velocity advection : It is written :

$$u_t + \vec{V}.\vec{\nabla}u = 0 \ \ in \ \ \mathbb{R}^2 \ \ with \ \ \vec{V} \epsilon \mathbb{R}^2$$

A standard upwind discretization is the following :

$$area(C_i)(u_i^{n+1} - u_i^n) = -\Delta t \sum_{j \ neighbor \ of \ i} \alpha_{ij}(\theta_{ij} u_i^n + (1 - \theta_{ij}) u_j^n)$$

with

$$\alpha_{ij} = \eta_x^{ij} V^x + \eta_y^{ij} V^y$$

above

$$\theta_{ij} = \frac{1}{2}(sign(\alpha_{ij}) + 1)$$

and where $\vec{\eta}^{ij}$ is defined below.

LEMMA 1 :

The above 2-D advection scheme satisfies the Maximum Principle if, for every cell C_i, the following inequality holds :

$$area(C_i) - \Delta t \int_{\partial C_i^+} \vec{V}.\vec{\nu} d\sigma \geq 0.$$

where ∂C_i^+ denotes the part of ∂C_i where $\vec{V}.\vec{\nu}$ is positive.

(2) - Non-constant velocity :

This case has to be studied in a conservative formulation :

$$u_t + div(\vec{V}u) = 0.$$

where $\vec{V}$ is given but not constant, $\vec{V} = \vec{V}(x, y)$.

It is reasonable to consider that a numerical scheme which approximates this conservation law is stable if it preserves the positiveness of the dependent variable.

The conservative scheme is derived :

$$area(C_i)(u_i^{n+1} - u_i^n) = -\Delta t \sum_{j \ neighbor \ of \ i} \alpha_{ij}(\theta_{ij} u_i^n + (1 - \theta_{ij}) u_j^n)$$

with

$$\alpha_{ij} = \eta_x^{ij}(\frac{V_i^x + V_j^x}{2}) + \eta_y^{ij}(\frac{V_i^y + V_j^y}{2})$$

$$\theta_{ij} = \frac{1}{2}(sign(\alpha_{ij}) + 1)$$

LEMMA 2 :

The above scheme preserves the positiveness of its solution if the following inequality holds for every cell C_i :

$$area(C_i) - \Delta t \max_{\text{j neighbor of i}} \|\vec{\Gamma}_j\| \int_{\partial C_i} d\sigma \geq 0.$$

Application to the Euler scheme :

The following " reference time-step " will be computed on each cell :

$$\Delta t_i = area(C_i)/(\lambda^i_{max} \int_{\partial C_i} d\sigma)$$

with

$$\lambda^i_{Max} = Max(\lambda_i, \max_{\text{j neighbor of i}} \lambda_j)$$

and

$$\lambda_i = (u_i^2 + v_i^2)^{\frac{1}{2}} + c_i$$

where u_i, v_i, c_i hold for the values in cell C_i of (resp.) horizontal velocity, vertical velocity, and sound speed.

In practice, time-step larger than Δt_i by a factor of 3 can be used (L^2-)stably and a good strategy for multi-gridding is to set Δt in the range of $2.5\Delta t_i$ to $3.\Delta t_i$.

3. MULTIGRID SCHEME

3.1. *Basic Iteration Method*

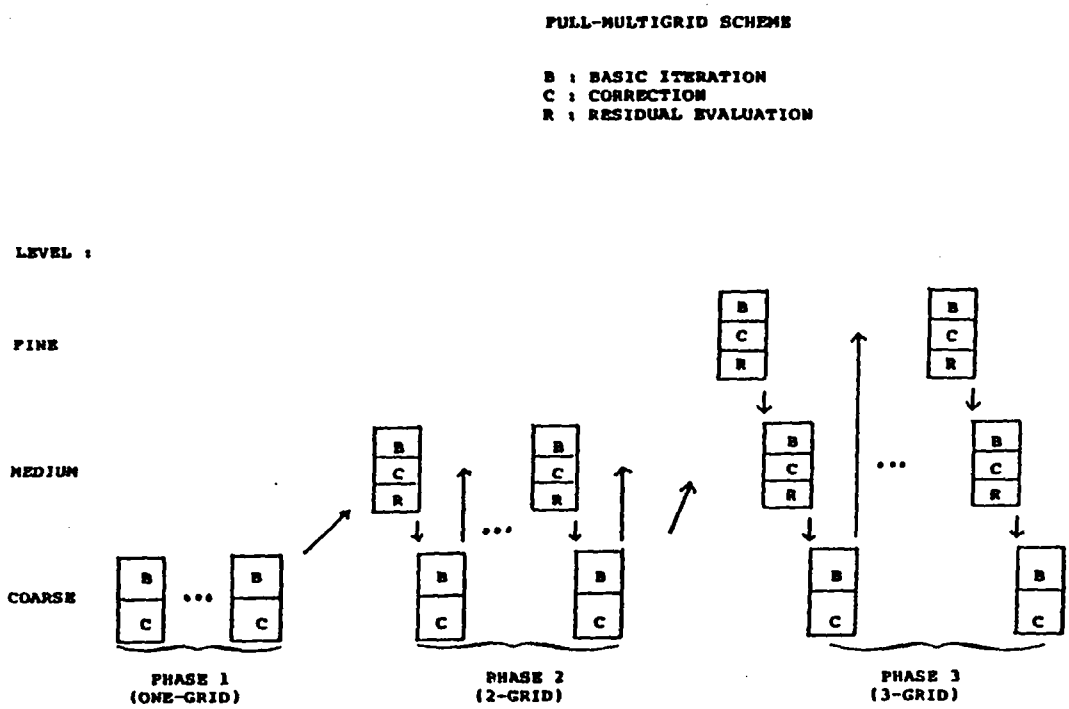
Following A. JAMESON [JAM1], a Runge-Kutta scheme is applied with either one or four time-steps ; in the second option, the following coefficients are employed (see [TUR], [LAL2]) :

$$\alpha_1 = .11$$
$$\alpha_2 = .2766$$
$$\alpha_3 = .5$$
$$\alpha_4 = 1.$$

3.2. *FMG first-order scheme*

The basic algorithm uses FAS iterations and Full Multigridding as in [JAM1]. We have it sketched for 3 grids. The transfer operators are defined in the present approach as follows :

- Fine-to-coarse : values are averaged in a conservative manner.
- Coarse-to-fine : the trivial injection is applied.



3.3. *Second-order version*

The second-order spatial scheme is introduced into the fine-grid solver only for the last phase of the full-multigrid process. This introduces a minor modification in the algorithm. However, two disadvantages appear in this construction : first the

coarse level correction is less consistent with the fine level smoother ; second, in a full-multigrid approach, the third phase starts from a first-order (medium level) solution instead of a second-order one. However, it will be seen in the sequel, that this does not result in a too severe convergence degradation.

4. NUMERICAL ILLUSTRATION

4.1. *Two experiments with nested meshes*

The calculations of an internal flow in a channel with a 4.2 % thick bump are presented. It has been observed that the regime defined by a Mach number at infinity equal to .85 is representative of the usual stiffness of transonic flow calculation.

A first mesh is presented in Fig.1 and contains 161 vertices. The three successive levels are also depicted : the dual fine level (161 control volumes), the medium level, the coarse level. The convergence histories are shown for - standard one-grid calculations with each of the three levels (the initial data correspond to uniform flow), - a one-grid calculation over the successive levels using as initial data the result obtained from the previously employed coarser grid, - a full-multi-grid calculation, that is one-grid scheme with the coarse level, then two-grid scheme with the medium one, and three-grid with the fine triangulation.

In order to evaluate the behavior of the scheme when the number of nodes is increased, we present the same experiment (Fig.2a to 2h) with a finer triangulation derived from the previous one by dividing equally each triangle into four new ones. The triangulation now contains 585 vertices. However again only three grids are employed. A comparison of the convergence history proves that the convergence rate is rather constant in each phase, and then approximately equal to the coarse-grid convergence.

Finally, we present a first-order four-grid FMG calculation. In the fourth phase, the residual is reduced by three orders of magnitude after 155 cycles (see Fig. 2j). The fourth coarse level is shown in Fig. 2i.

4.2. *Application to a strictly non-structured mesh*

The efficiency is now evaluated with the calculation of an external flow around a NACA0012 airfoil (Mach =.72, angle of attack = 0 deg.). The triangulation is now really unstructured : it results from the use of a mesh generator based on an element front algorithm and contains 800 vertices (Fig.3a). The convergence is again fast ; the second order accurate solution is obtained in about 150 three-grid cycles in the third phase when the one-step Runge-Kutta is applied (Fig.3f) while

the convergence of the first-order version required only 80 cycles (Fig.3e). The loss of symmetry in the coarse grids does not seem severely penalizing. The Mach and pressure contours of the resulting solution are shown in Fig.4.

4.3. *Comparison with a classical approach*

A second important experiment is the comparison with (what we call) **a multi-triangulation algorithm** : three triangulations are nested standardly (by element division) and as spatial scheme, the second-order upwind scheme is applied at each level. The successive nested triangulations contain respectively 121, 442, and 1684 vertices.

We present three calculations relying on the fine triangulation, using **four-step Runge-Kutta** three-grid algorithms and starting from uniform flow.

A comparison of the convergence histories of the two algorithms is presented in Fig.5e :
- when the multi-triangulation algorithm is applied with second order flux-splitting over the three levels, the solution is obtained in about 40 cycles.
- when the same algorithm is applied, but with first order flux-splitting over the two coarse levels and the second order splitting over the fine level, the solution is obtained also in about 80 cycles, with a more monotone convergence.
- with the presented algorithm, the solution is obtained in about 80 cycles (with the same convergence).

This seems to prove that the difference between these two approaches essentially comes from the lack of accuracy of coarse grid smoothers. For heavier calculation, the asymptotic convergence speed can be a more important factor and the two approaches may have comparable efficiency.

4.4. *Application to a locally refined mesh*

The combination of local mesh refinement and multigrid algorithm is frequently advocated ; successive grid levels are constructed by **local** refinement of the previous grid level. One disadvantage of this approach is that these levels operate only **locally** and this may reduce the speed-up(that is the ratio of efficiency between 1-grid and multigrid algorithms) with respect to the standard global multigridding, since in some region the 1-G scheme is applied.

In this section, we wish to demonstrate that the coarsening / agglomerating algorithm enables us to generate **global** coarse levels, in order to keep the complete multigrid speed-up.

We start from a locally refined mesh, constructed for the calculation of the

flow past a cylinder [18]. The fine mesh contains 2141 nodes ; then a medium mesh is derived, containing 598 zones, and finally a coarse one with 244 zones. The ratio of the levels is satisfactory. In Fig. 6a,b,c,d. the different levels are shown to demonstrate the regularity of the partitions. To compare algorithms we considered the case of a freestream Mach number of 0.38. It appears from Fig. 6e. that applying the second-order method (over the fine level only) compared to applying the first-order method over all levels results in a reduction of convergence rate (in terms of iterations) by a factor noticeably less than 2.

4.5. *3-D calculations*

A few 3-D experiments have been performed. The flux splitting used is now van Leer's [LEE2]. Firstly, we compute the flow in a channel with 4.2 % bump with a tetrahedrization which is essentially a 72x21x3 mesh. The Mach at infinity is .85. The first-order accurate mesh is used. We present in Fig. 2. the history of the convergence when a Full four-grid algorithm is applied ; it can be (favourably) compared to the 2-D 3-grid multi-triangulation calculation in [LAL4], see Fig. 7.

Two-jet flow in chamber : A practical illustration of the efficiency of the 3-D code is the calculation of two impinging supersonic jets in a rocket combustion chamber. The first scheme is applied. Starting from a uniform condition in the chamber, the convergence to steady-state is prohibitively low when an explicit iteration is applied. On the contrary, the convergence in Full Multi-Grid mode is very fast (Fig. 8).

This is a typical case of stiffer flow, that requires a longer time to be steady, as for example blunt body or airfoil flows at high angles of attack [FEZ3], the external-internal flow around and in an air intake [ANG8].

4.6. *Conclusion*

The explicit MG scheme is an interesting alternative to explicit time stepping : the gain in efficiency for 1000 - 2000 nodes in 2-D is around 2-3 ; this is less than the gain obtained with the implicit solvers described in chapter II, for which the factor is always better than 4. An open question arises : would these two approches be combined ?

5. IMPLICIT MG SCHEME

The subject of this section is the inclusion of a MG algorithm in the linear phase of the unfactored linearized implicit 2-D scheme described in Chapter II. This approach

has been already studied by Mulder [MUL] and Schroder-Hanel [SCH] ; it is less ambitious than the fully nonlinear scheme of Hemker and co-workers [HEM]. In choosing this first step, we want firstly to upgrade an implicit code the avantageous for its a satisfactory behavior in a large range of Mach numbers (from Mach = 10^{-3}, see [BEN], to Mach = 25, see [STO3]), and secondly, to get informations related to the linear case before studying the nonlinear one.

Furthermore, it must be noted that this approach combines two algorithms whose convergence rates are essentially mesh independent; indeed the time-iteration is near of Newton's method when only first-order approximations are applied in both explicit and implicit phase; for the case of the first-order preconditioning of a second-order approximation we refer to the discussion of chapter II.

The single grid scheme is the low-storage one described in chapter II. In the MG version of this scheme, the linear system is solved using a MG algorithm : basic iteration is again the collective Jacobi iteration ; the residuals are transferred by same transfer operators as for the explicit scheme ; but now the usual V-cycle (instead of the saw-tooth-cycle) is applied.

5.1. *Numerical experiments.*

We present only a few measures and an illustrative experiment : the test case is again the flow in a channel at Mach equal to .85 ; the mesh contains 2225 points. Two typical convergence histories one-grid, six-grid are presented in Fig. 9. The cost of the construction of the zones is less than the cost of one single grid iteration ; the cost of one linear cycle is about 1.5 the cost of the corresponding single-grid iteration, as predicted by the theory. The overall gain in efficiency is about 1.8. However, before giving a precise evaluation, the optimization of the MG code has still to be done.

For a more close relation to theoretically predicted behavior, we next consider the ideal 2-Grid scheme that consists in a 2-Grid scheme in which full convergence is realized on the coarse grid. Considering only the linear convergence in the implicit phase of the first time-step, we note that mesh independency is rather well satisfied (Fig.10). This property is unfortunately not as well satisfied in the global nonlinear process (Fig.11). We conjecture that for heavier meshes, the nonlinear time-stepping would behave better so that a still larger efficiency would be attained.

We finally present two calculations for verification of the conjecture of chapter II : the calculations with no angle of attack are those of the internal transonic flow in a channel over a 4.2 % thick bump, a GAMM Workshop test case, and the external transonic flow over a NACA0012 airfoil at zero degree of incidence. In both cases the freestream Mach number is 0.85. The timestep Δt is increased as the convergence

proceeds according to the following empirical rules giving the C.F.L. number:

$$C.F.L. = \begin{cases} \max\left(10^3, 10/RES\right) & \text{in case (a)} \\ \max\left(10n^3, 10/RES\right) & \text{in case (b)} \end{cases}$$

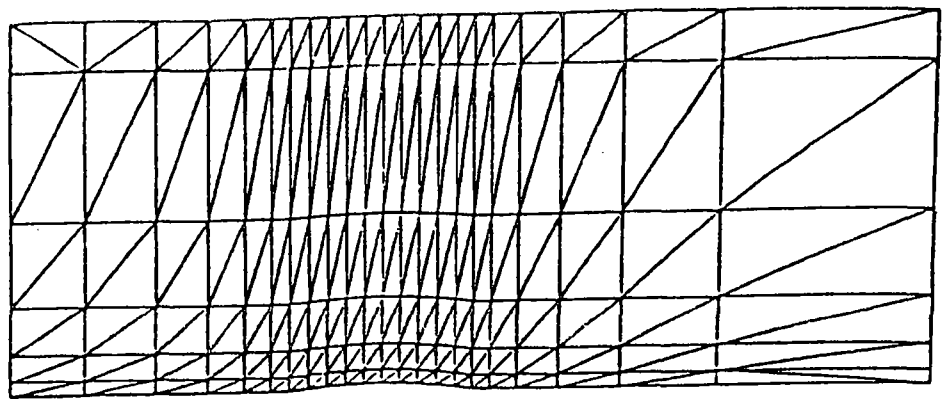
in which "RES" denotes the L2-norm of the explicit right-hand side normalized to the initial value and n the iteration counter. The inversion of the system is made at each iteration by a 5-level multigrid algorithm. The convergence history is shown on Figure 12, where it appears that after some initial phase during which transient phenomena are observed, the asymptotic rate predicted in chapter II is achieved by the iterative method.

Consequently, in these computations, the required number of timesteps for the residual to be reduced by a prescribed factor is independent of meshsize (provided the timestep can be made large enough stably); since, in addition, the computational work involved in each inversion of the implicit system by multigrid is essentially proportional to the meshsize, so is also the computational work of the entire calculation.

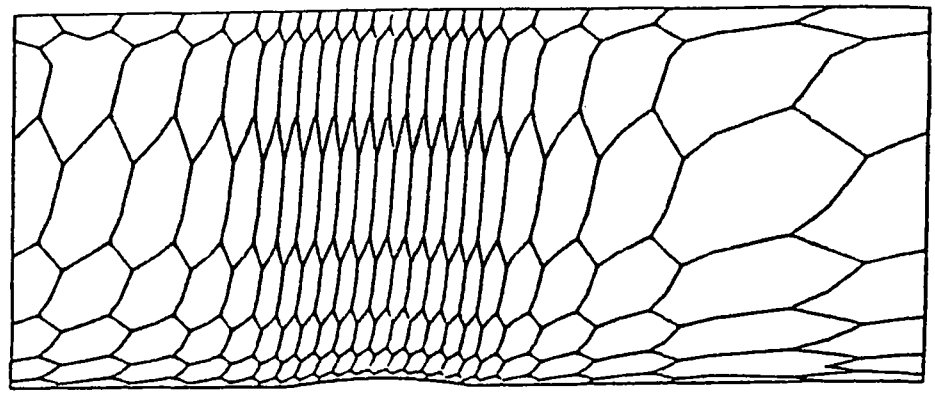
6. NOTES :

This chapter describes essentially two chapters of the Ph. thesis of M-H. Lallemand [LAL1] and is taken in part from [LAL3] for the explicit 2-D scheme, [FEZ5] for the implicit one. The 3-D version have been developed by H. Steve [STE1]. Recently a different way to construct a second-order solver has been derived by Koren and Lallemand, using the Defect Correction idea [paper in preparation].

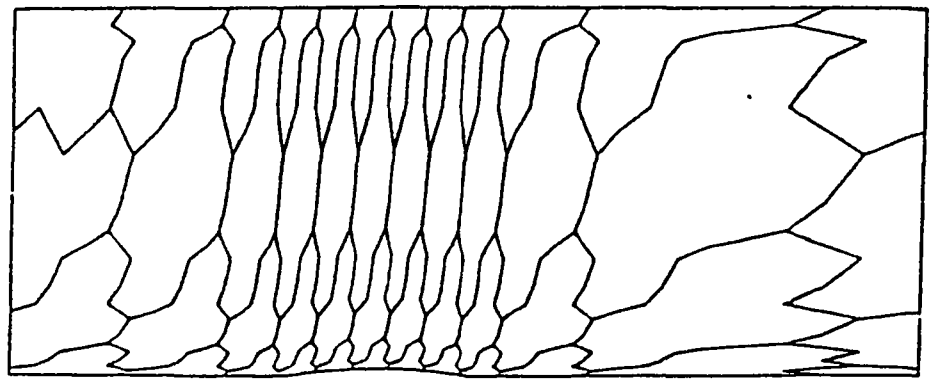
1a



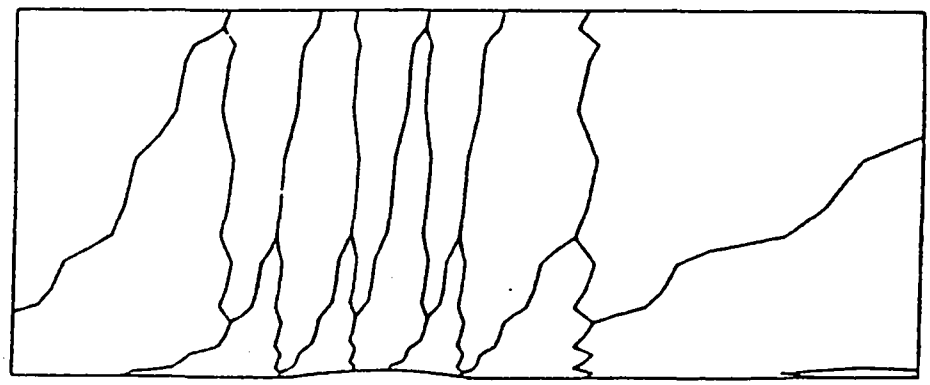
1b



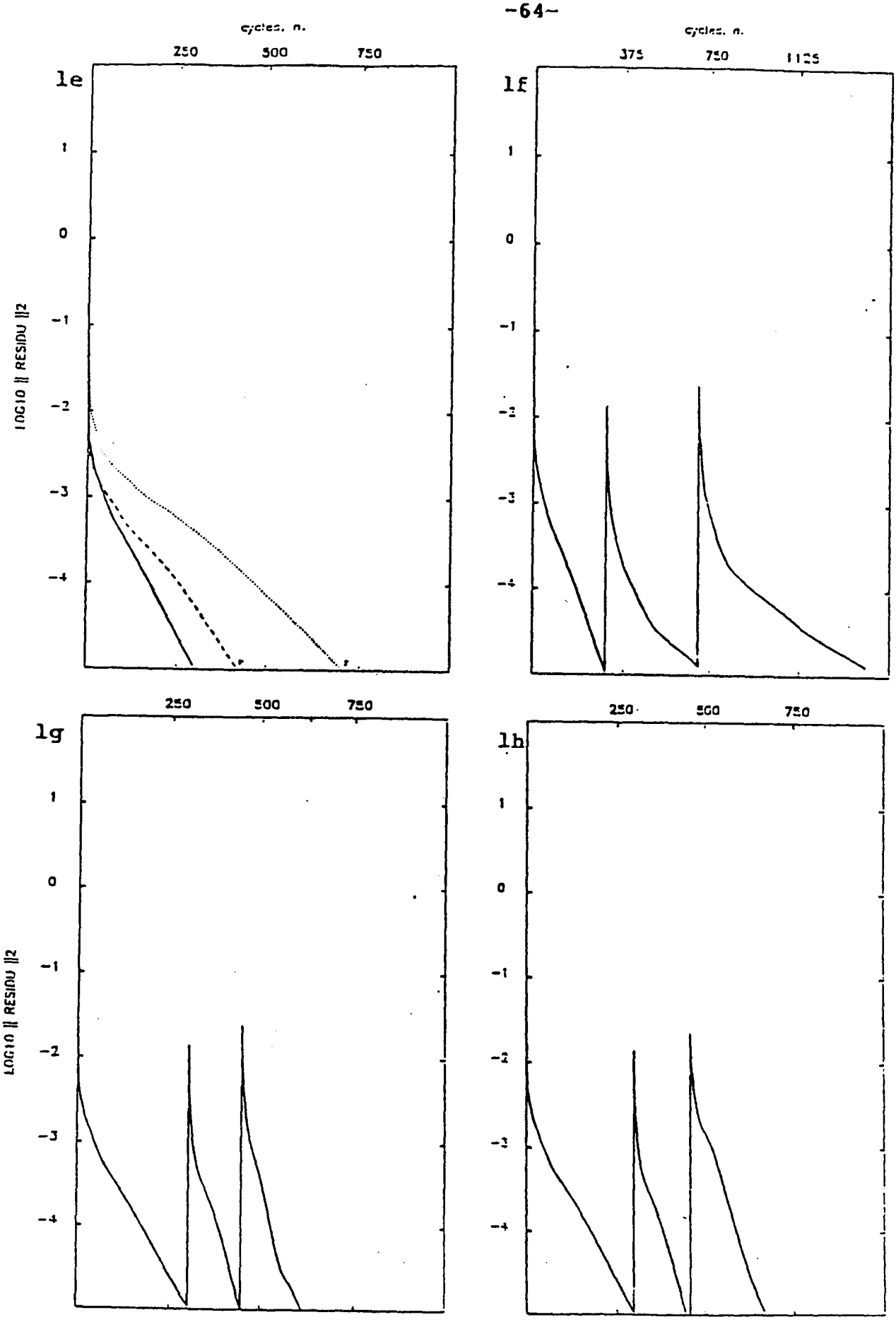
1c



1d

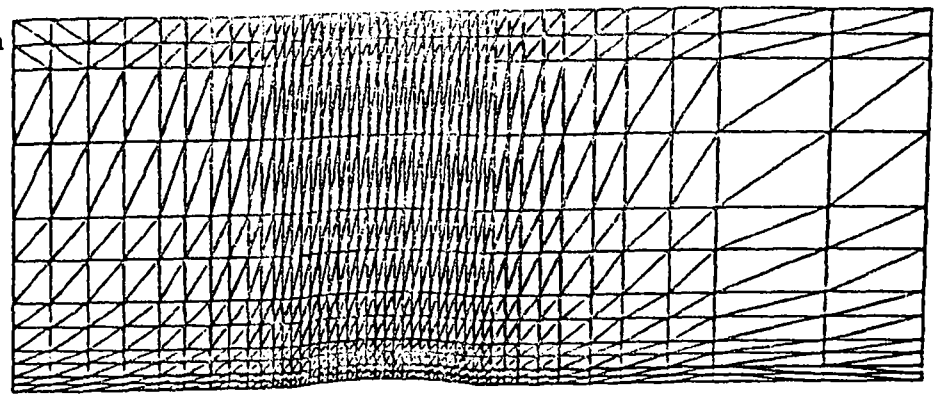


Figures 1a to 1d : Flow in a channel with a 4.2% thick circular bump. Mach at infinity = .85 ; 1a : triangulation ; 1b : dual mesh ; 1c : medium level ; 1d : coarse level.

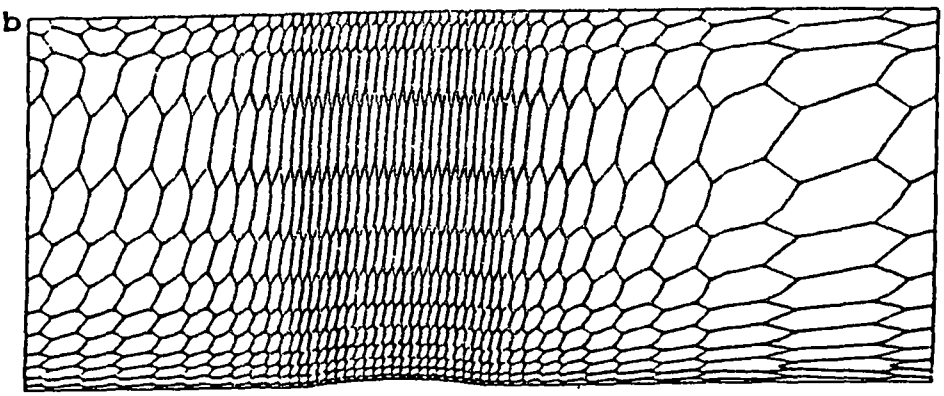


Figures 1e to 1h : Convergences : 1e : one-grid with each grid ; 1f : one-grid with coarse level initial solution ; 1g : FMG, first-order ; 1h : FMG, second-order.

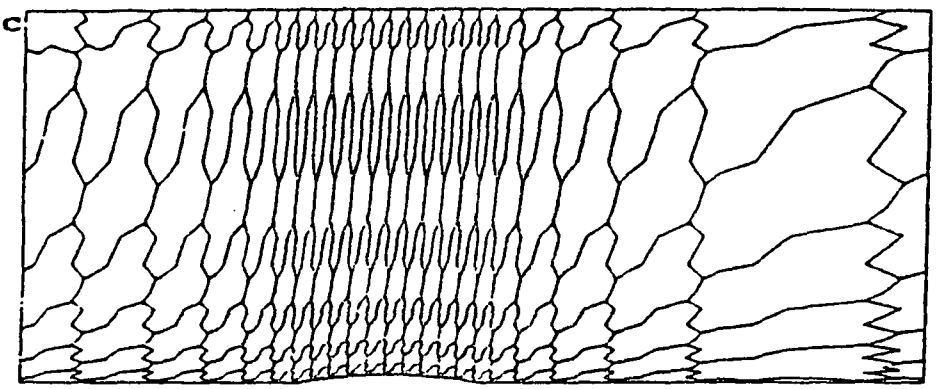
2a



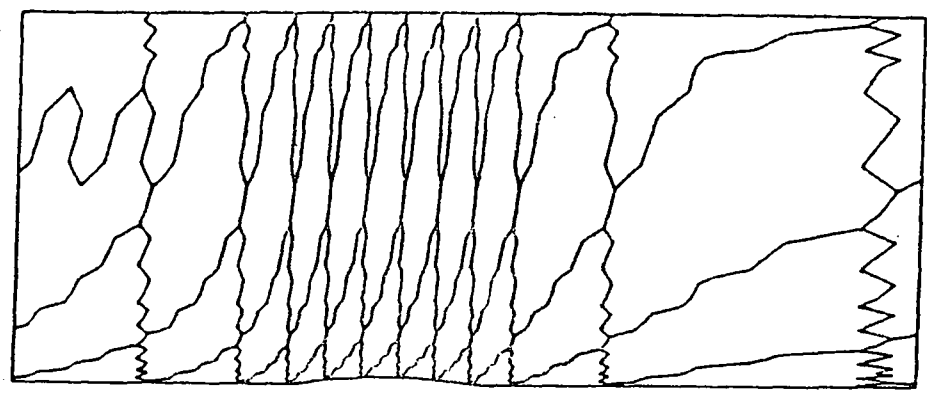
2b



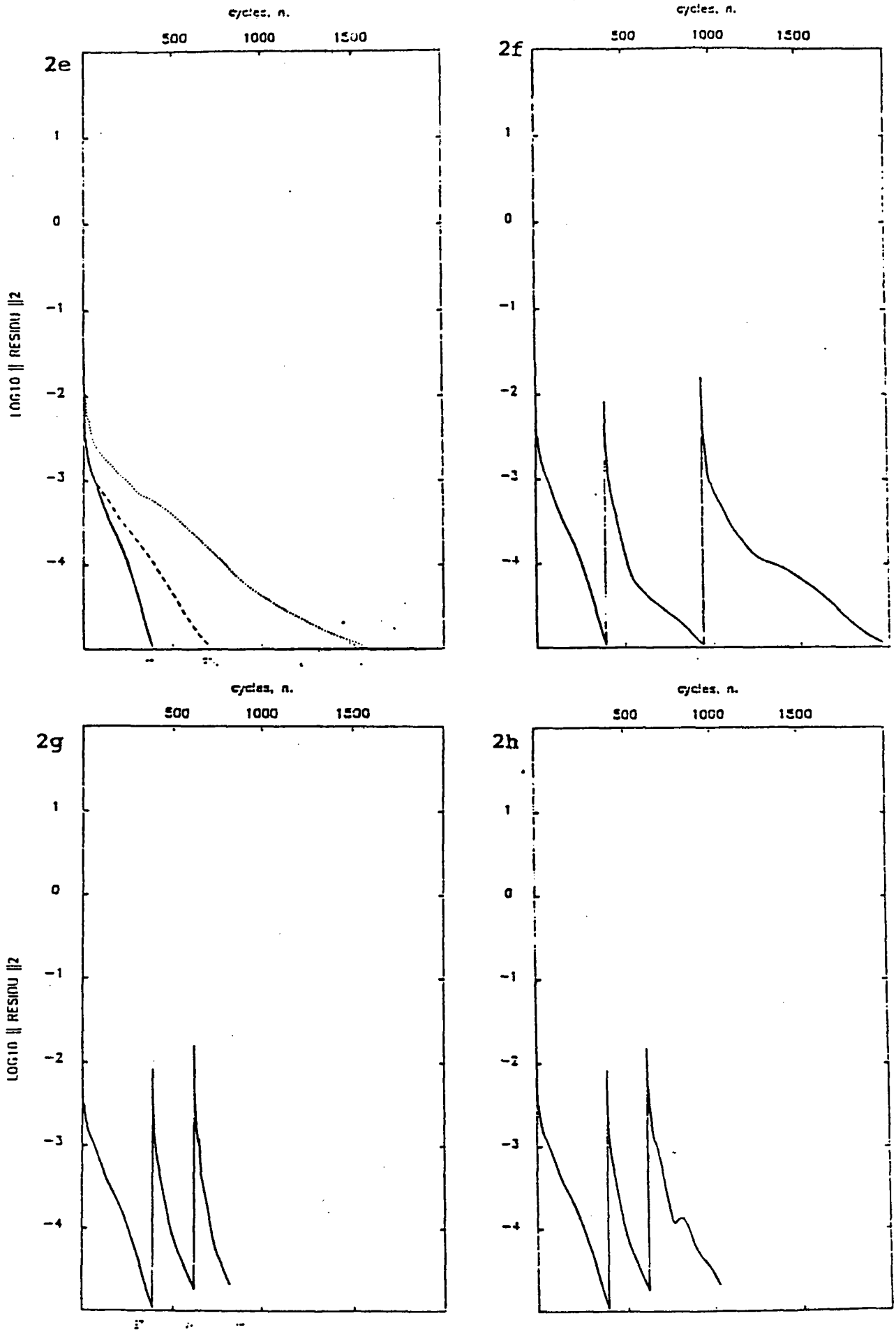
2c



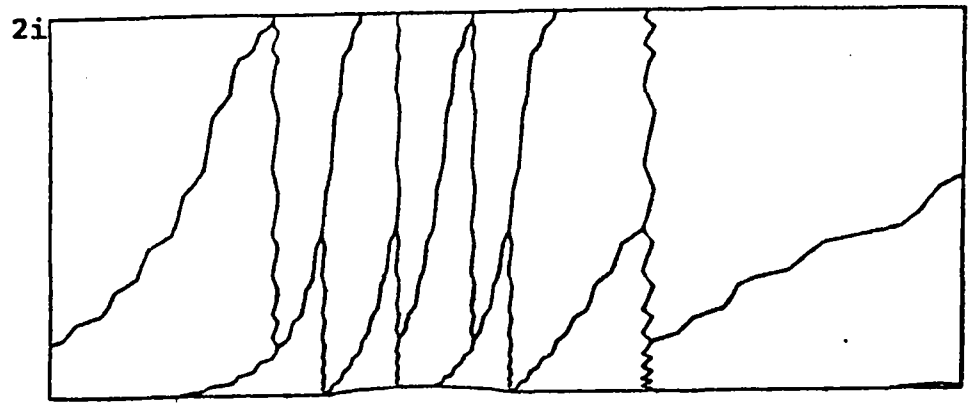
2d

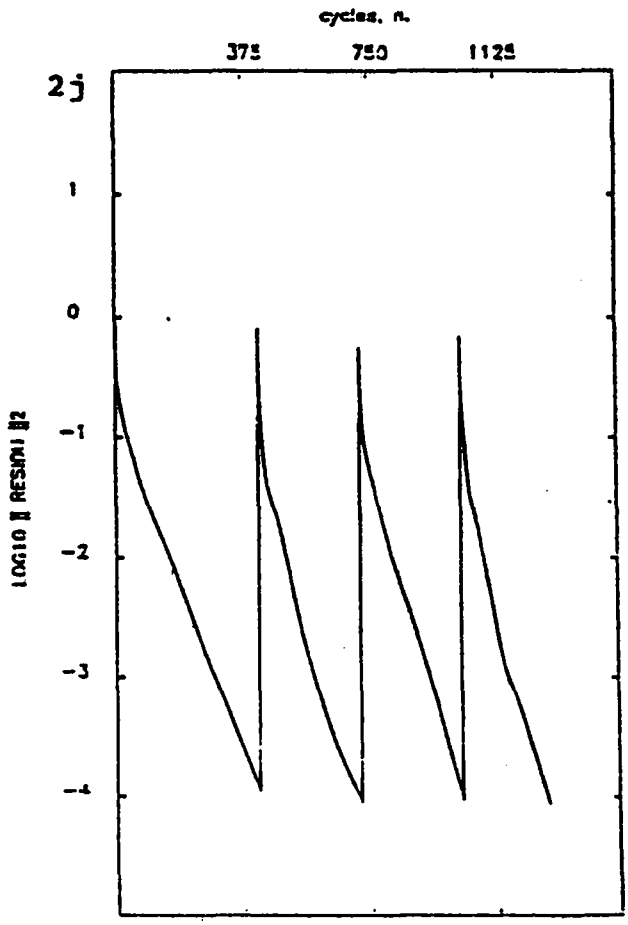


Figures 2a to 2d : Same as Fig. 1, with a finer triangulation.

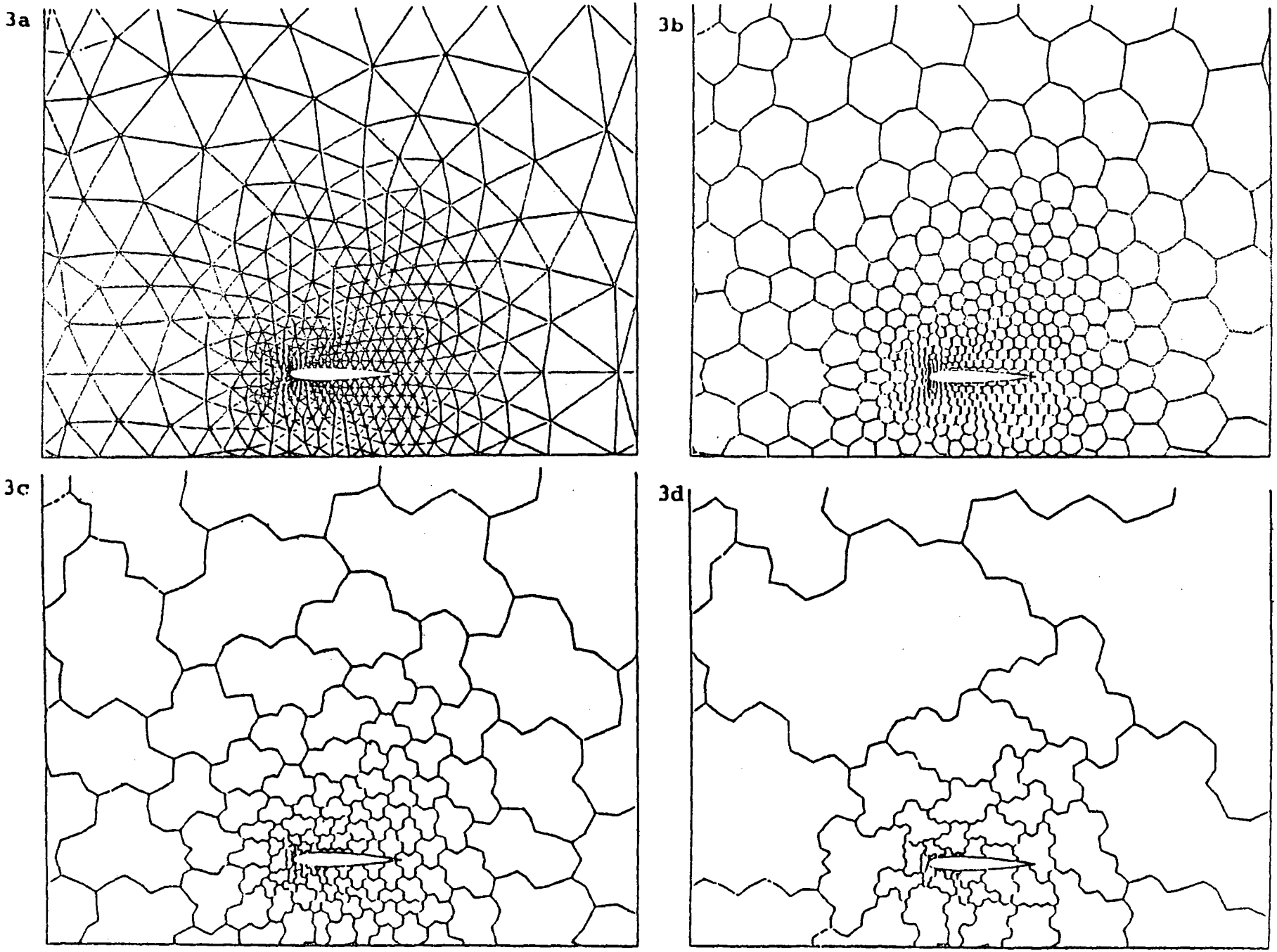


Figures 2e to 2h : Convergence.

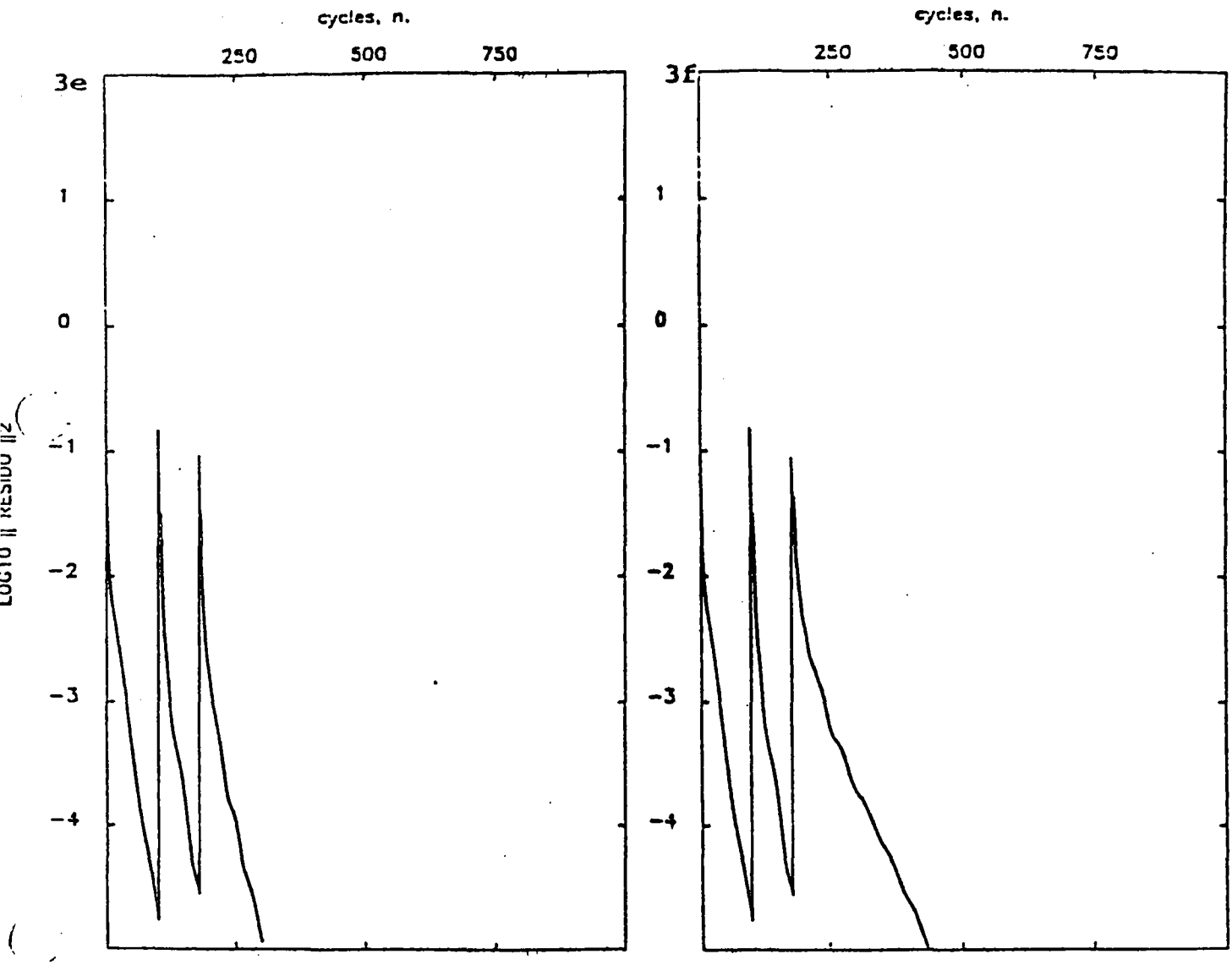




Figures 2i to 2j : Four-grid calculation ; 2i : next coarser grid (level 4) ; 2i : FMG - convergence.

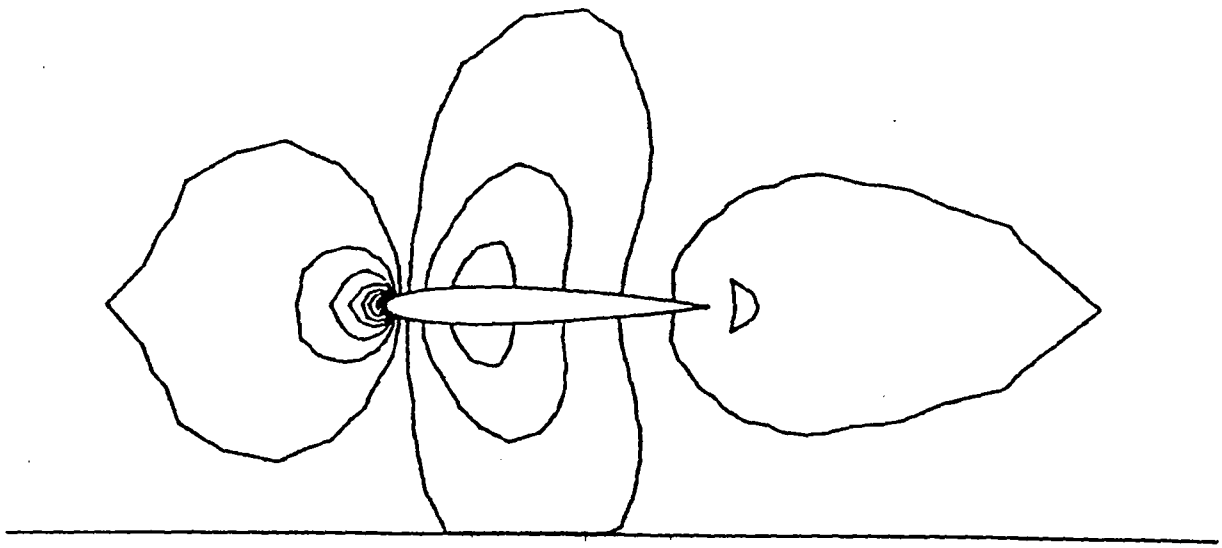


Figures 3a to 3d : Flow around a NACA 0012 airfoil.
Mach at infinity = .72, angle of attack = 0 deg. ; 3a : triangulation ; 3b : dual mesh ; 3c, 3d : coarser meshes.

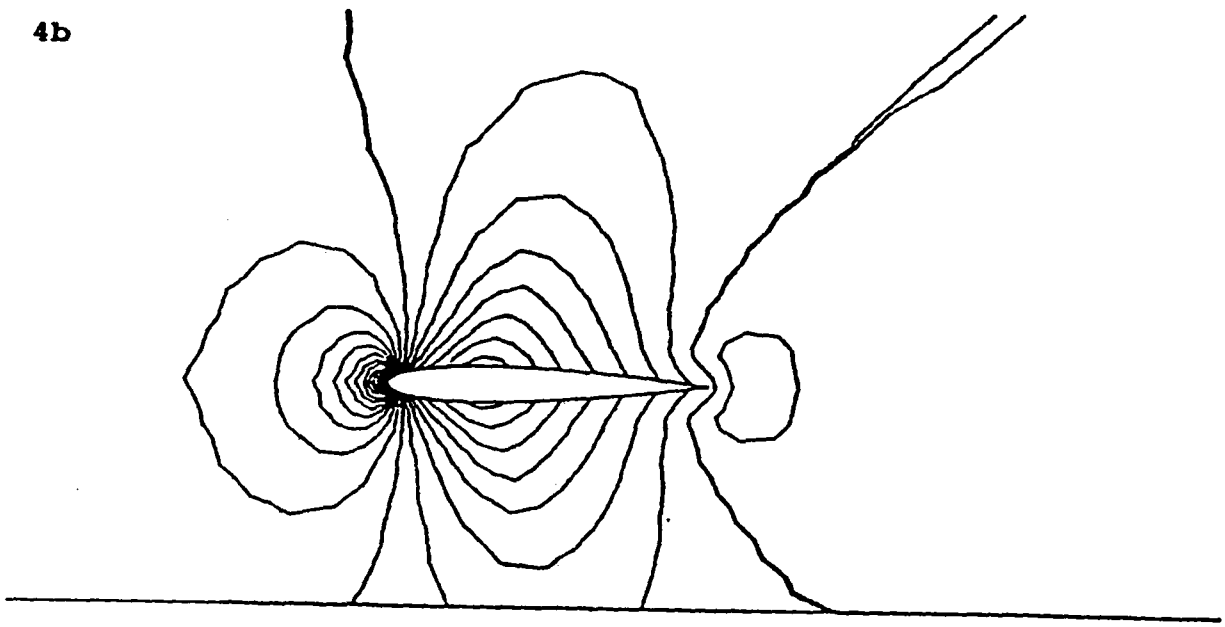


Figures 3e to 3f : 3e : FMG convergence with first-order scheme
3f : FMG, second-order.

4a

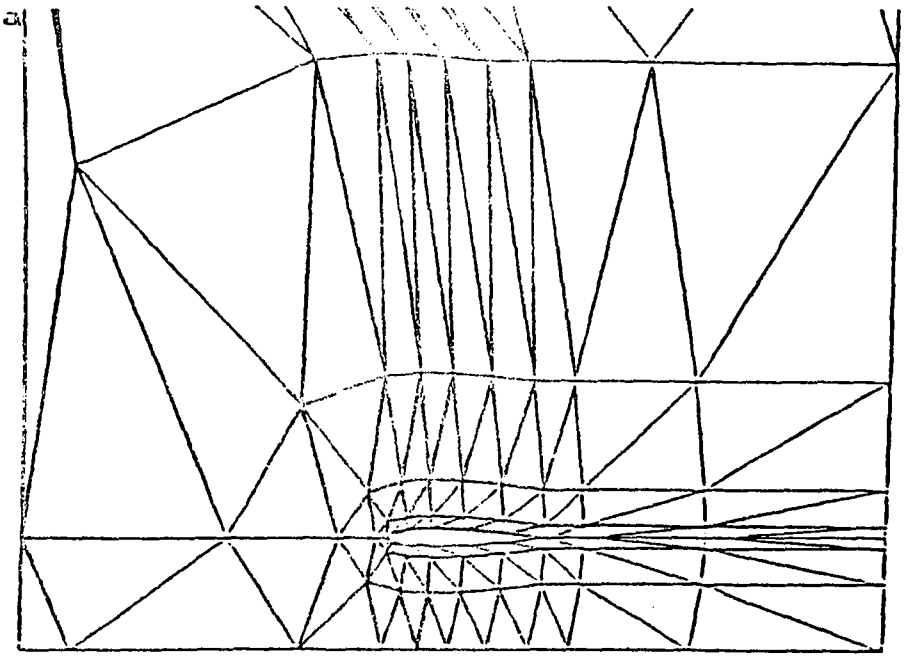


4b

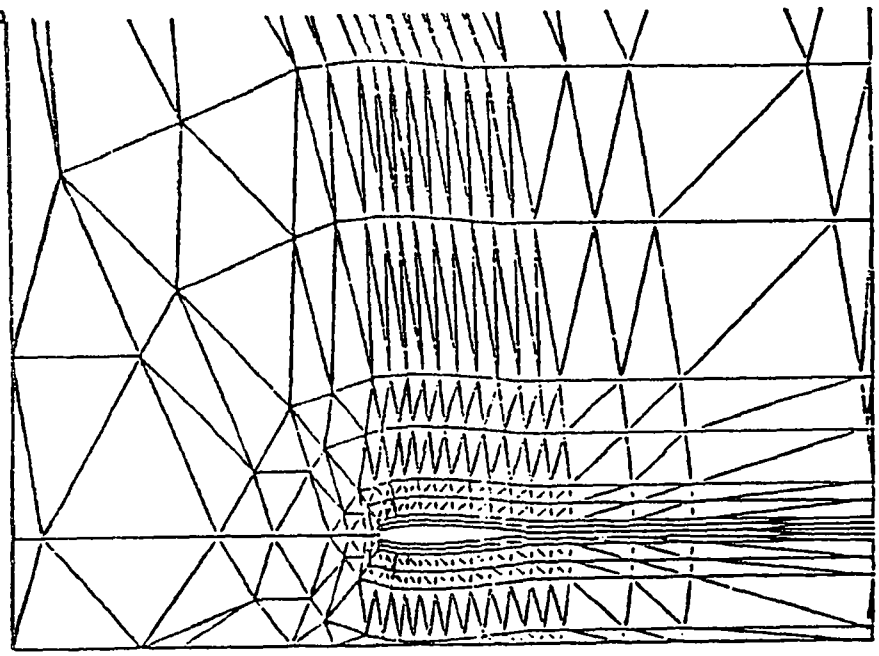


Figures 4a to 4b : Flow around a NACA 0012 airfoil, continued. Mach and pressure contours.

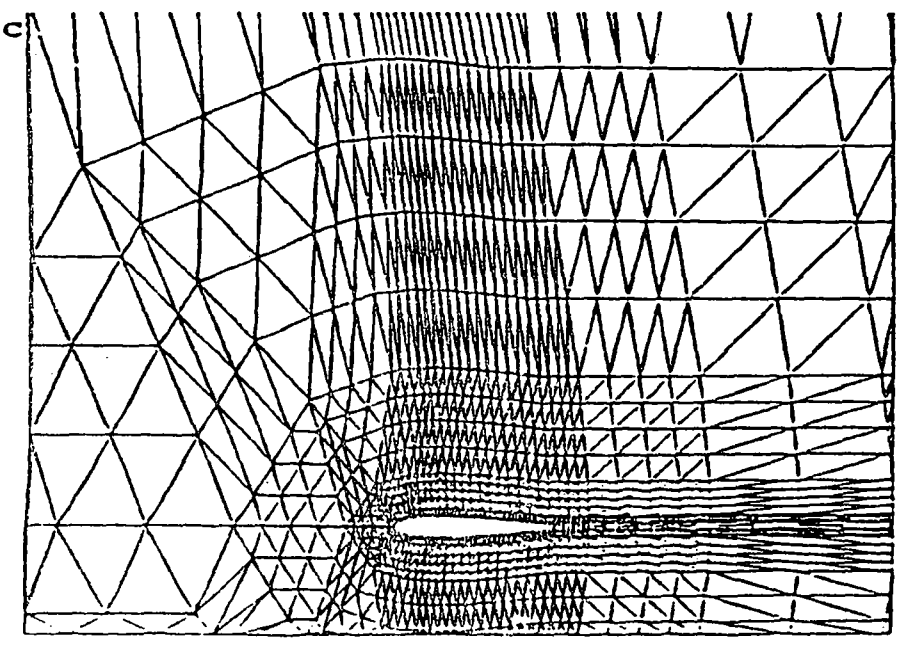
5a



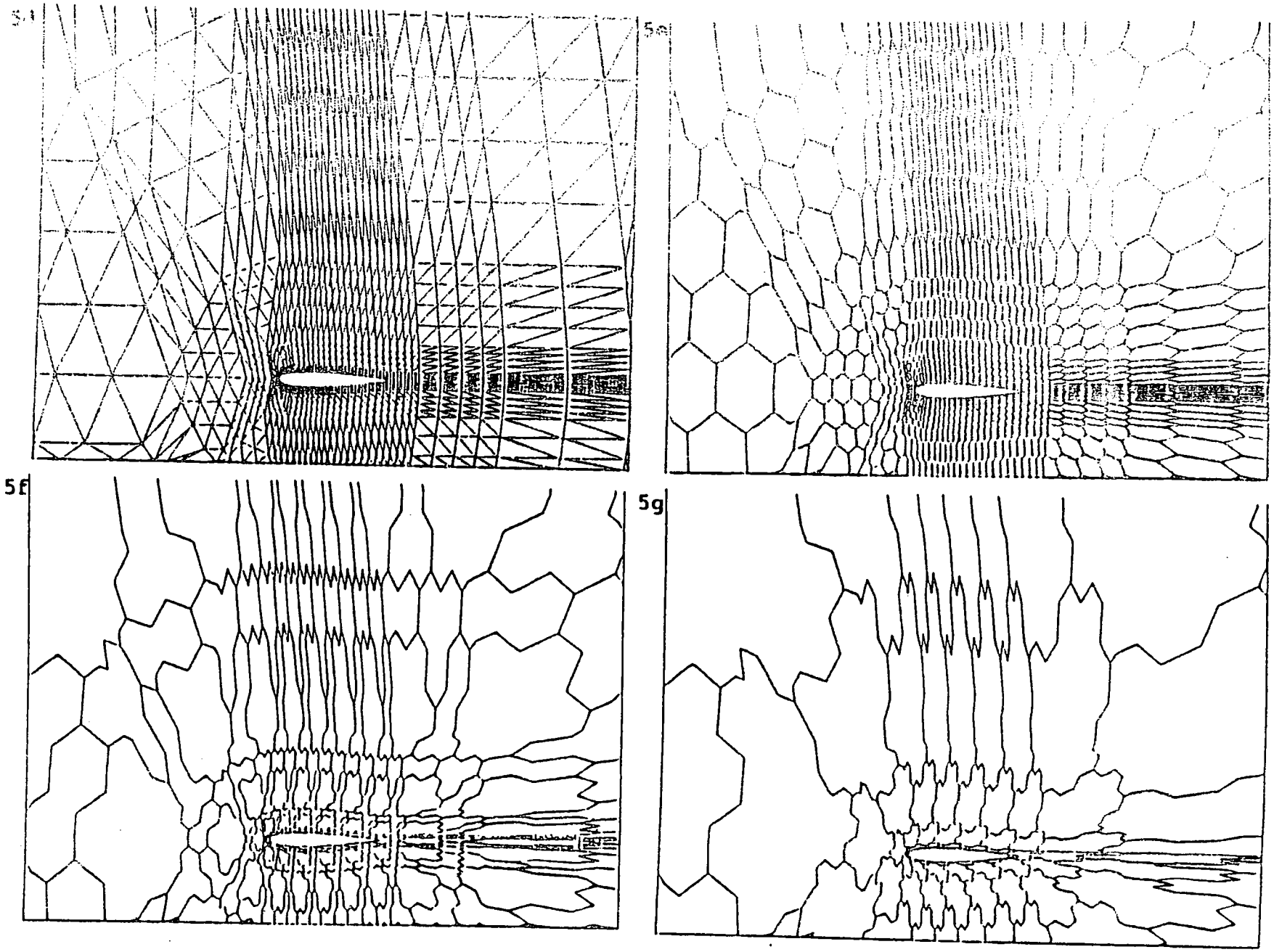
5b



5c



Figures 5a to 5c : Triangulations used in the multi-triangulation scheme.



Figures 5d to 5g : The levels used in the presented scheme.

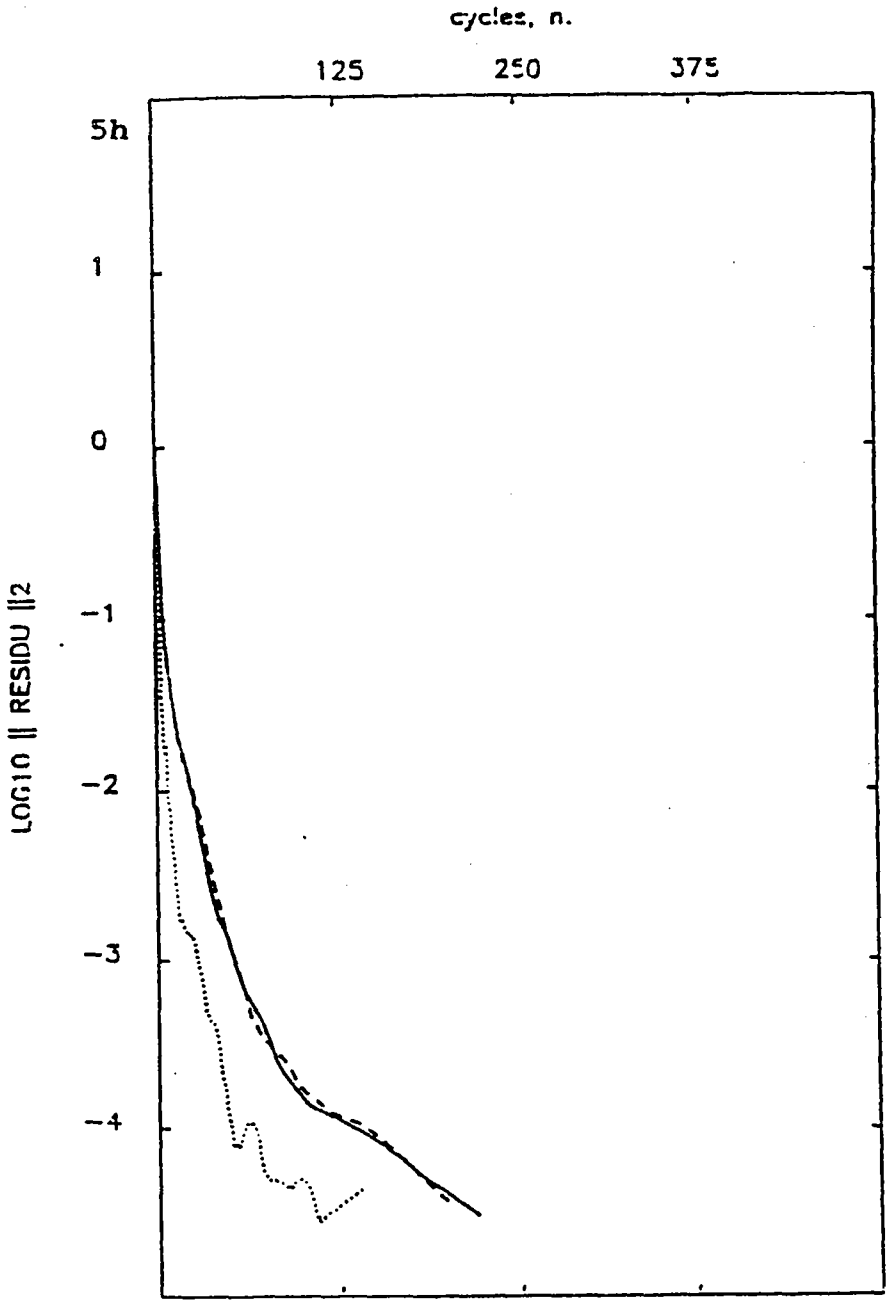


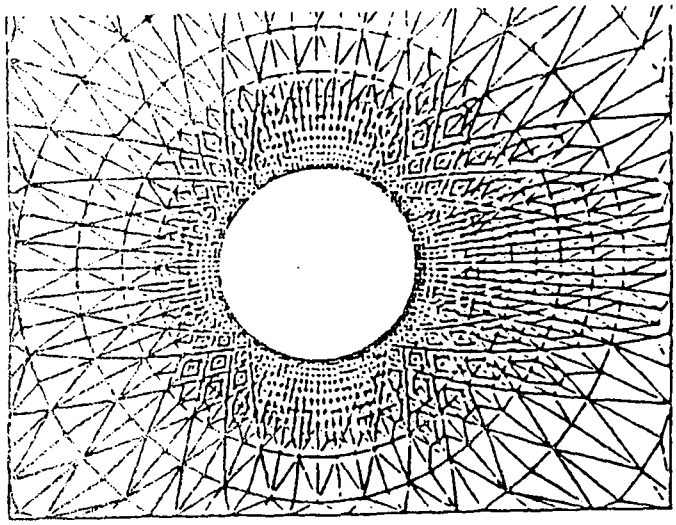
Figure 5h : Flow around NACA 0012 airfoil comparison with a multi-triangulation (MT) scheme (8,10).

... MT algorithm with 2nd-order scheme applied on each level.

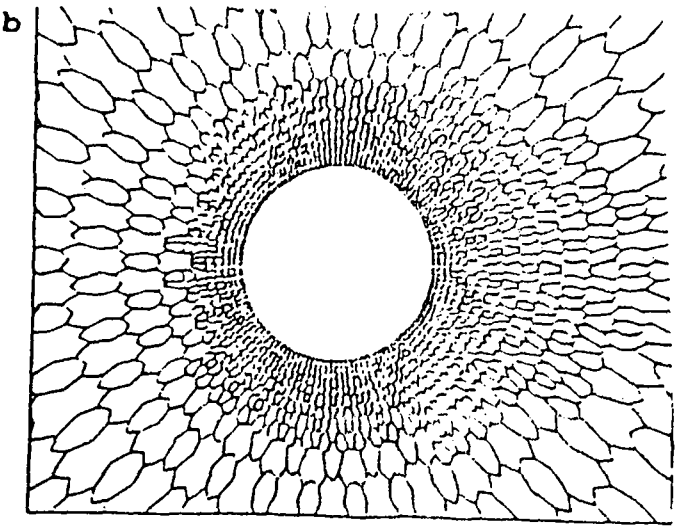
--- MT algorithm with 2nd-order scheme applied on the fine level only

___ presented 2nd-order version agglomerating algorithm.

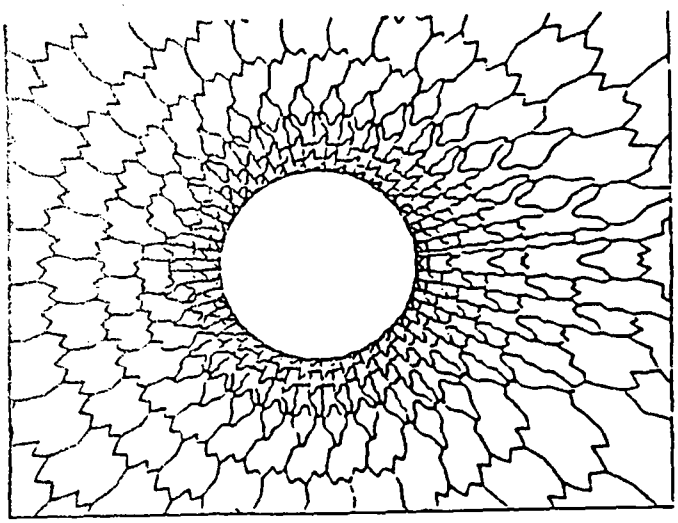
6a



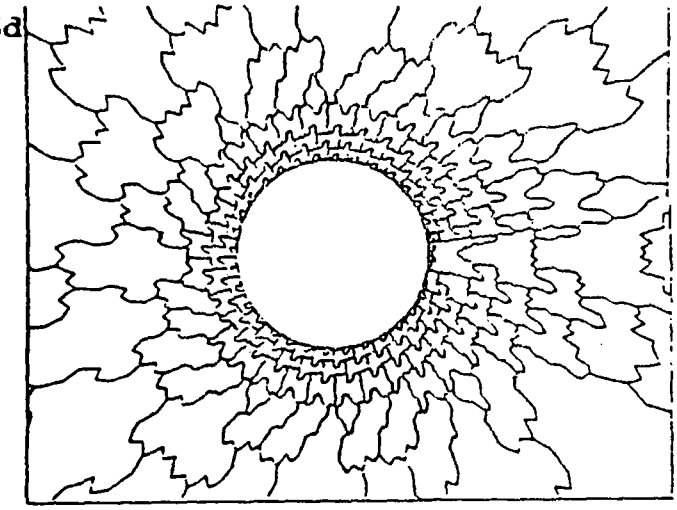
6b



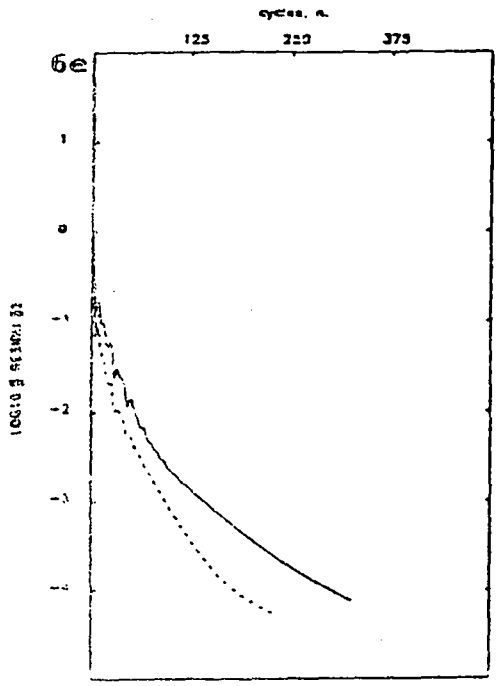
6c



6d



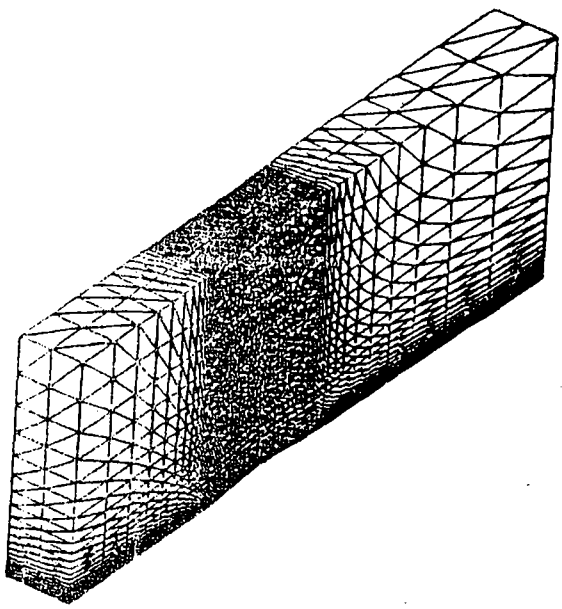
6e



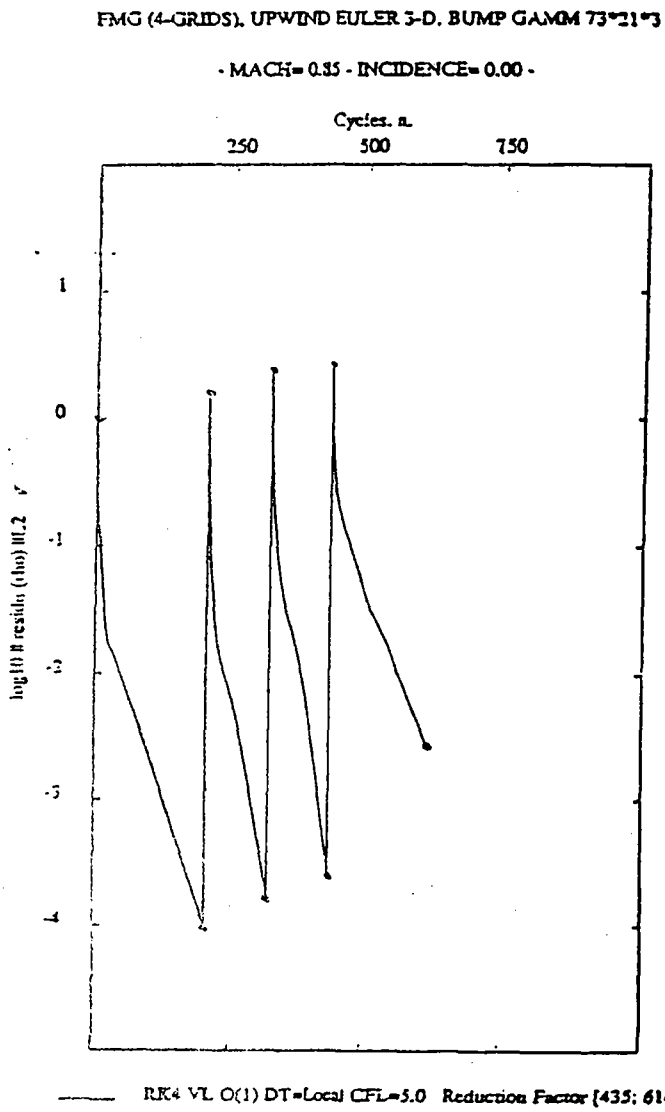
Figures 6a to 6e : Flow past a cylinder. Comparison between 1st-order and 2nd-order version of the algorithm.

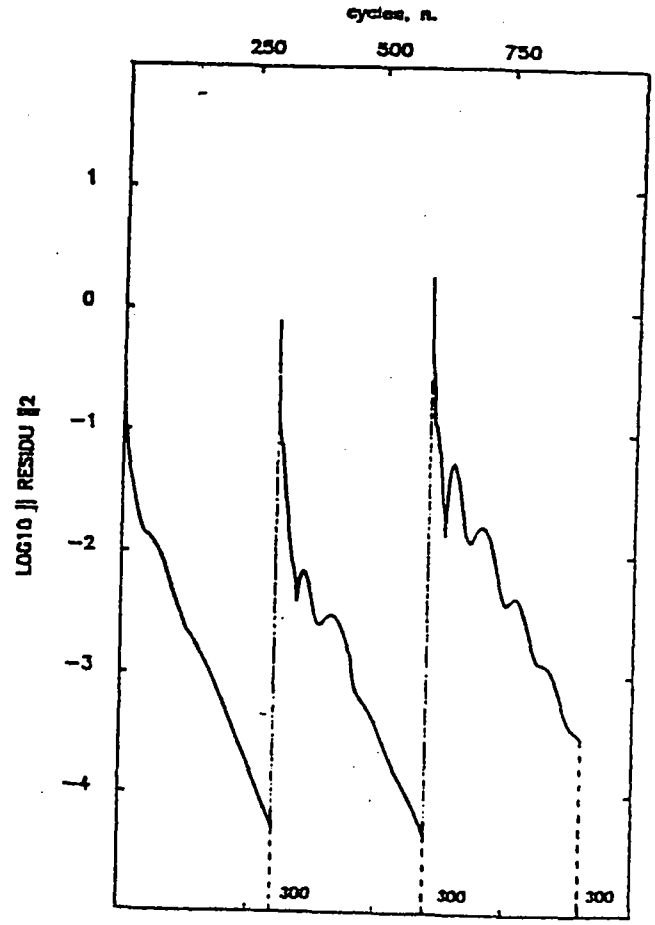
—— 2nd-order

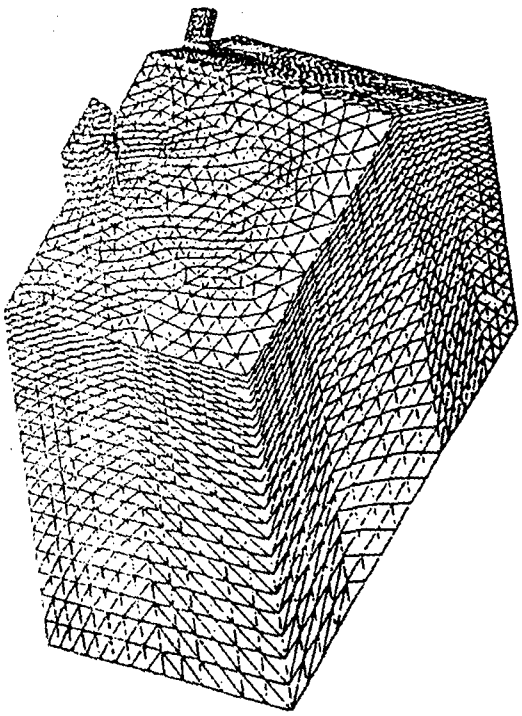
--- 1st-order.

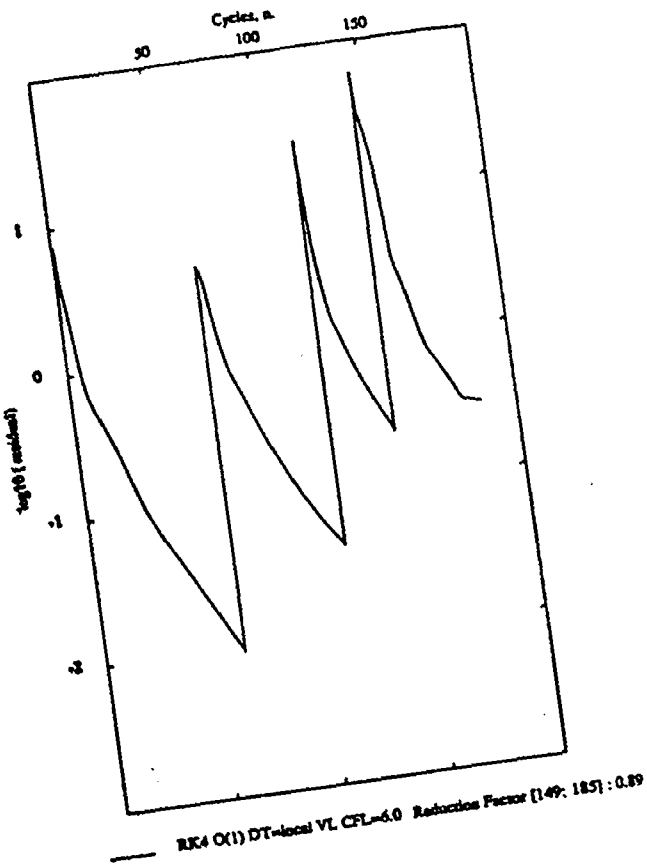


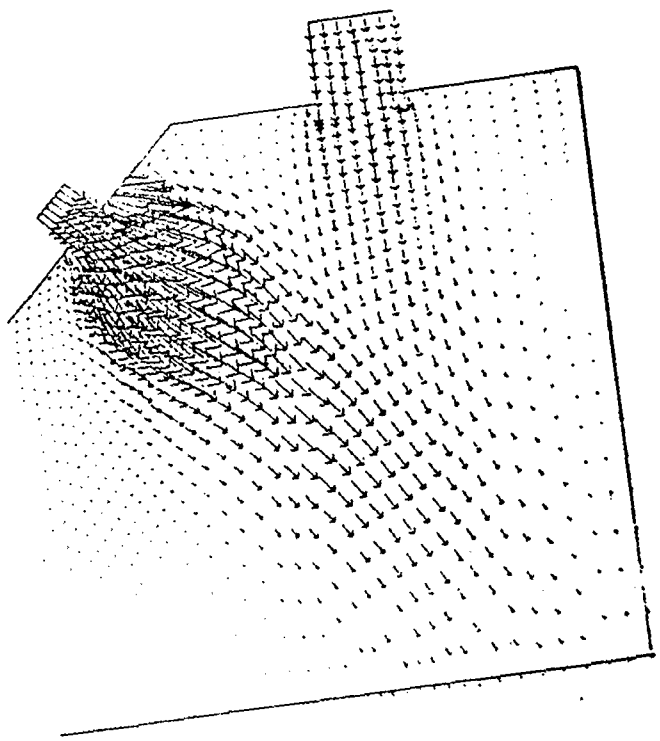
Figures 7a to 7c :
First-order 3-D scheme ;
a comparison of 3-D FMG (a,b)
with a 2-D multitriangulation
analog (c).











Figures 8a to 8c :
First-order 3-D scheme :
Transonic impinging jets in a
combustion chamber (21 000 nodes

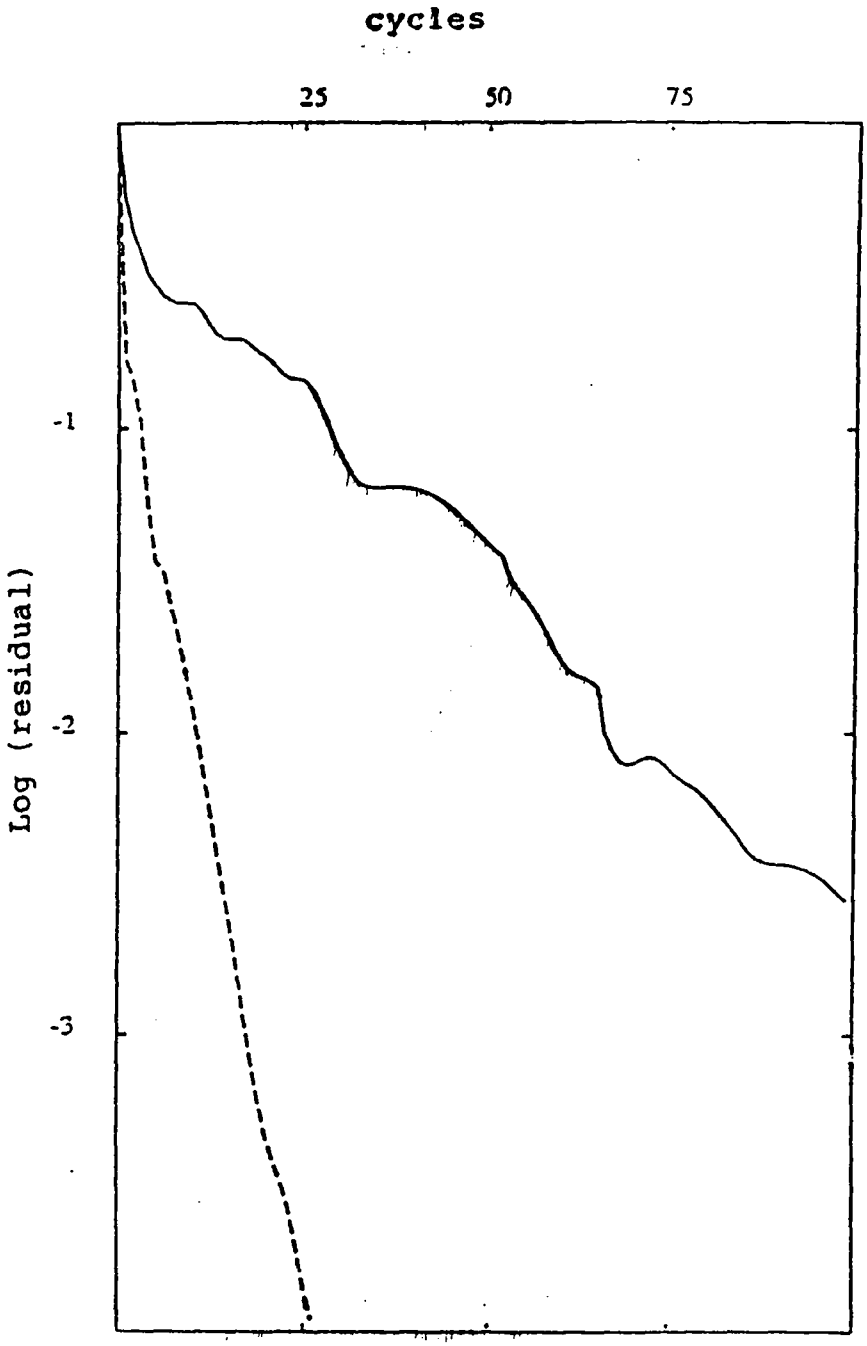


Figure 9 : Comparison of the multigrid implicit Scheme with the single-grid implicit one.

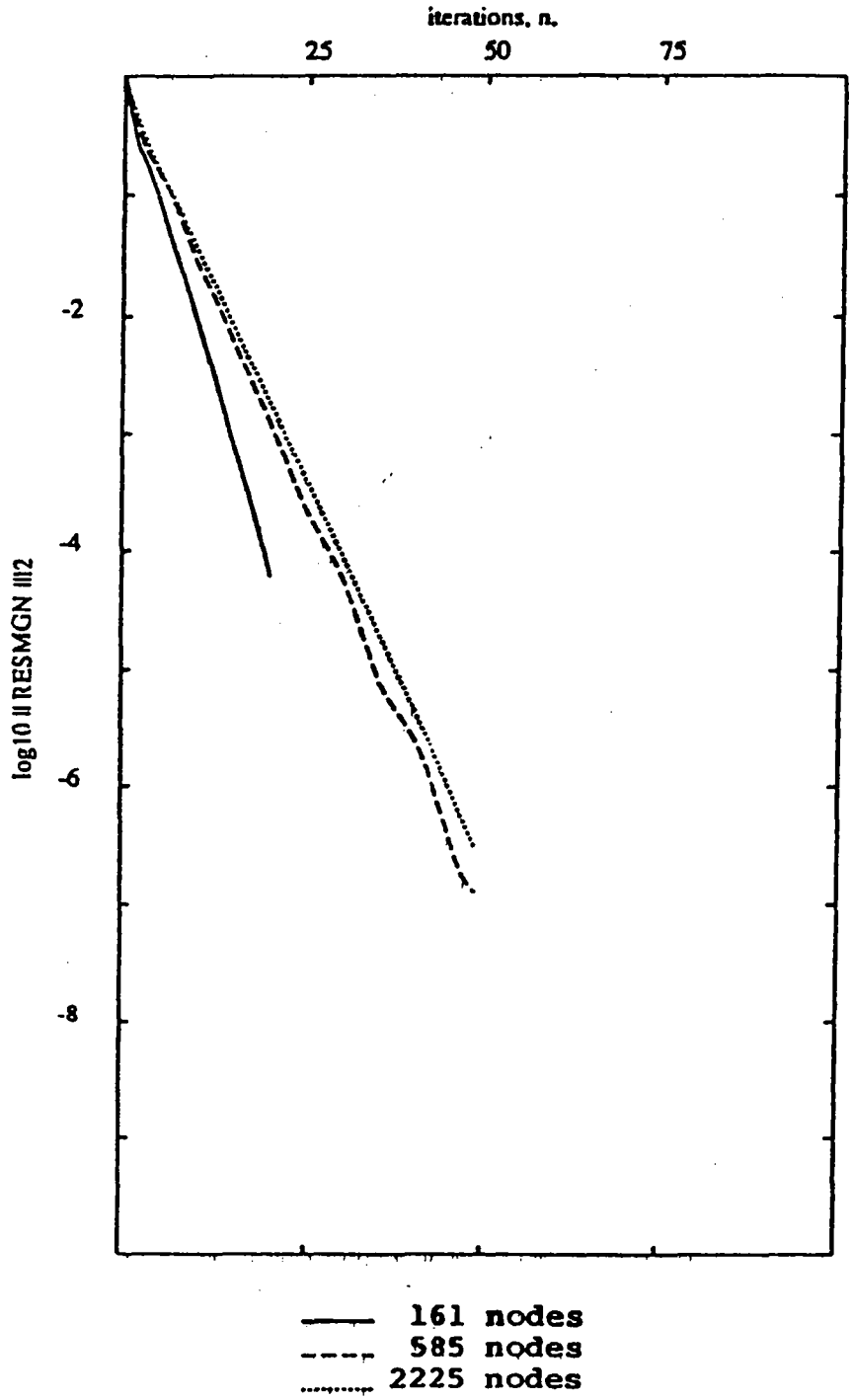


Figure 10 : Two-Grid scheme ; comparison of convergence in the linear phase for a sample of nested meshes.

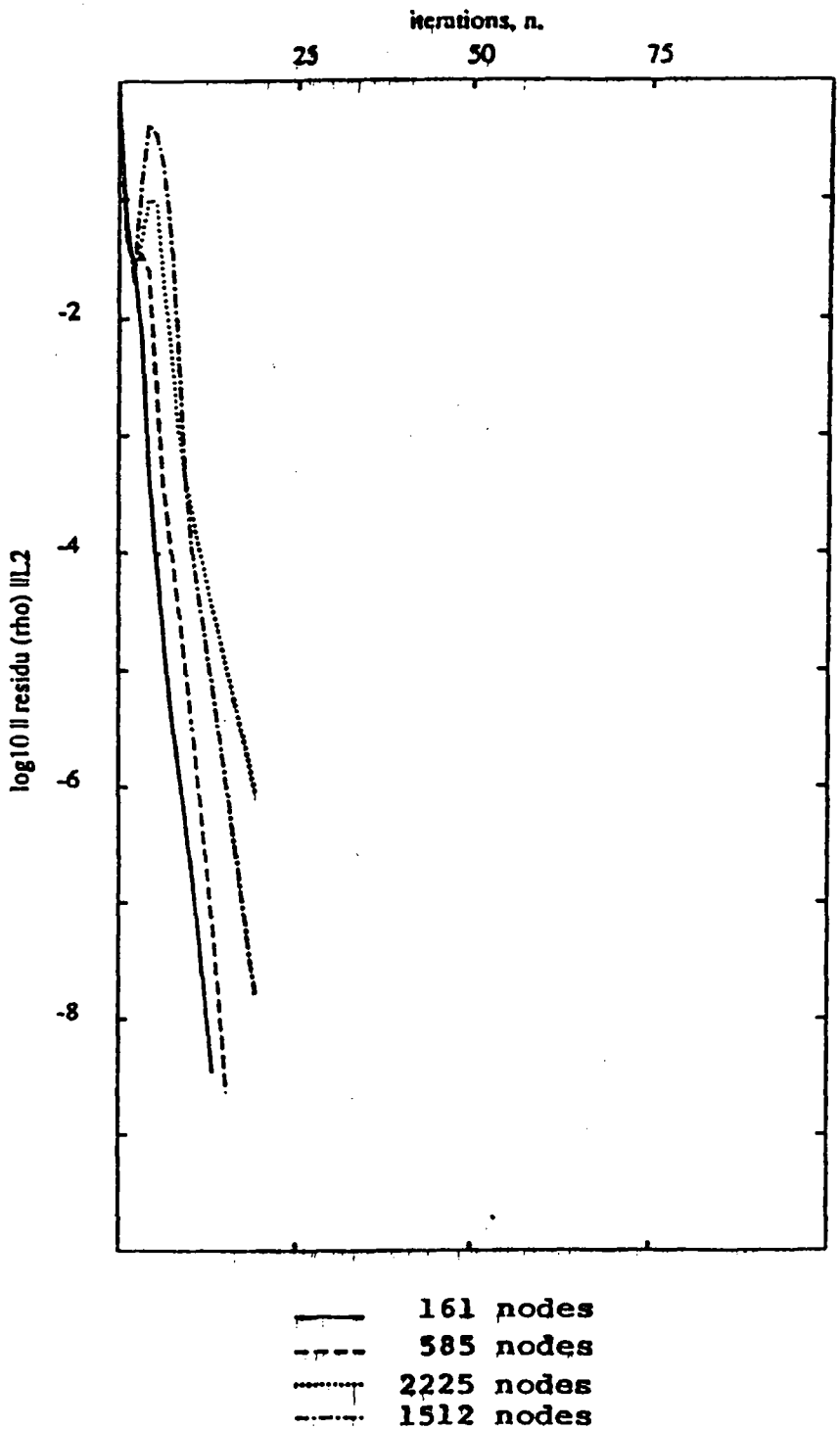


Figure 11 : Two-grid scheme, comparison of convergence in the global (non linear) time stepping

Finite-element computations of transonic flows
- Mach number = 0.85 - Incidence = 0.00 -

(a) internal flow in a channel over a bump

(b) external flow over a NACA0012 airfoil

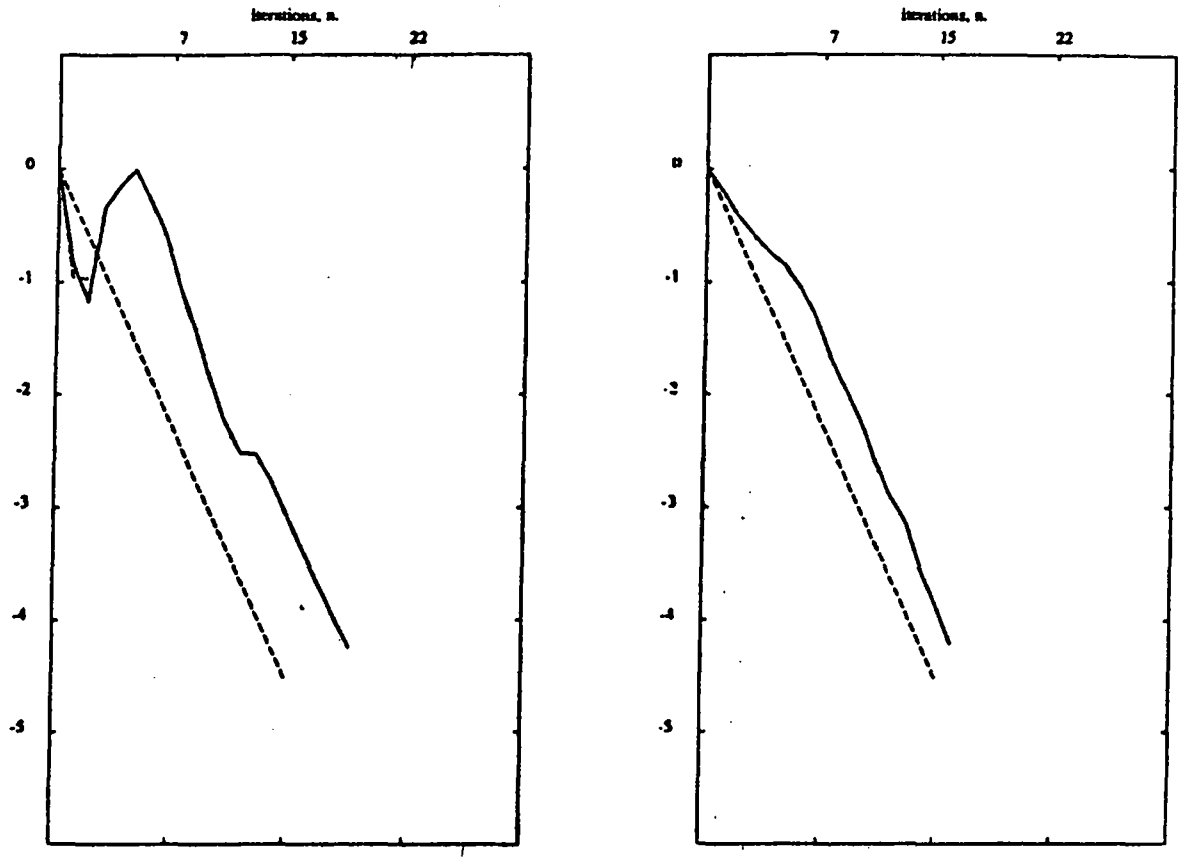


Figure 12: Convergence history of the implicit MG method based on $\frac{1}{2}$-upwind Osher's splitting explicitly, 1st-order Steger-Warming splitting implicitly, and 5-level multigrid for the implicit solution

CHAPTER IV : MESH ADAPTION ALGORITHMS

One of the most important problems in compressible CFD is the grid generation around complex geometries. In the particular case of the Euler equations, the PDE system is hyperbolic and it appears that the generation of adapted grids is a stiff problem, even in the context of Finite Element-type triangular unstructured grids. In order to identify accurately the essential structures of the flow, the grid should satisfy the following important conditions :
(i) the grid should account for the geometry, (body-fitted grids seem mandatory),
(ii) the grid should be fine enough in the region where the equations to be solved need accuracy.

To reach objectives (i), (ii), we suggest the following procedure :
- Step 1 : An initial coarse grid Φ_0 is generated, fulfilling only condition (i).
- Step 2 : Adaptive procedures are applied, that use intermediate simulations, in order to derive from the initial grid Φ_0 new grids Φ_k satisfying conditions (i) and (ii).

In this section we discuss several devices for the realization of step 2. Two families of adaptive procedures can be recognized :
In the enrichment procedure, the location of the nodes is fixed but their number is increased.
In the moving node procedure, the nodes are moved while their number remains constant.

1. SELF-ADAPTIVE MESH ENRICHMENT

We are interested in this section in the first method. For solving the Euler equations, this method has been recently illustrated by locally refined Finite Volume techniques combined or not with multigrids, see Usab and Murman [USA] and Berger and Jameson [BER] for the 2-D cases, and also Baker, Jameson, Vermeland [BAK2] in a 3-D context. Finite Element Methods have also been presented by Lohner [LOH2] and Palmerio [PAL1, PAL3] Angrand and Leyland ([ANG6]).

1.1. *Adaptive procedure*

1.1.1. Triangulation and degrees of freedom

We consider a 2-D F.E.M. type triangulation of the domain of computation Ω. This

means that a polygonal region Ω_h is constructed, which is a "good" approximation of Ω. Then a triangulation Φ_h of Ω_h is constructed. The basic assumption is that the mesh is conformal, that is, that the intersection of two elements is either nothing, or a vertex, or a slide or an element, as sketched in Figure 1.

FIGURE 1 :
Conformal and non conformal meshes

Taking into account non conformal meshes would lead to a complicated handling of special nodes in the complete solution algorithm.

As far as mesh refinement is concerned, this means that we shall be able to increase locally the number of elements without changing the data structure describing the mesh. From this point of view, the unstructured grids provide more flexibility than F.D.M./F.V.M. ones, for which a locally refined mesh is described in a different manner from the initial (I, J, K)-mesh (extra loop and matching interfaces must exist in the computer program).

1.1.2. Enrichment algorithm

1. Firstly a criterion is used, to decide which triangles require additional effort on accuracy. For these triangles new points are introduced, located at the middle of each side.

2. The identification of the sides to be divided in then derived in a second step.

3. In the third step, new triangles are constructed by considering in the old triangles the number of sides containing new points (see Fig. 2) ; three situations can appear :

- If there is only one point added, the former triangle is split into two new ones.

- If two points must be added three sub-triangles are created.

- With three new points (on each mid-side), then the triangle is divided into four sud-triangles.

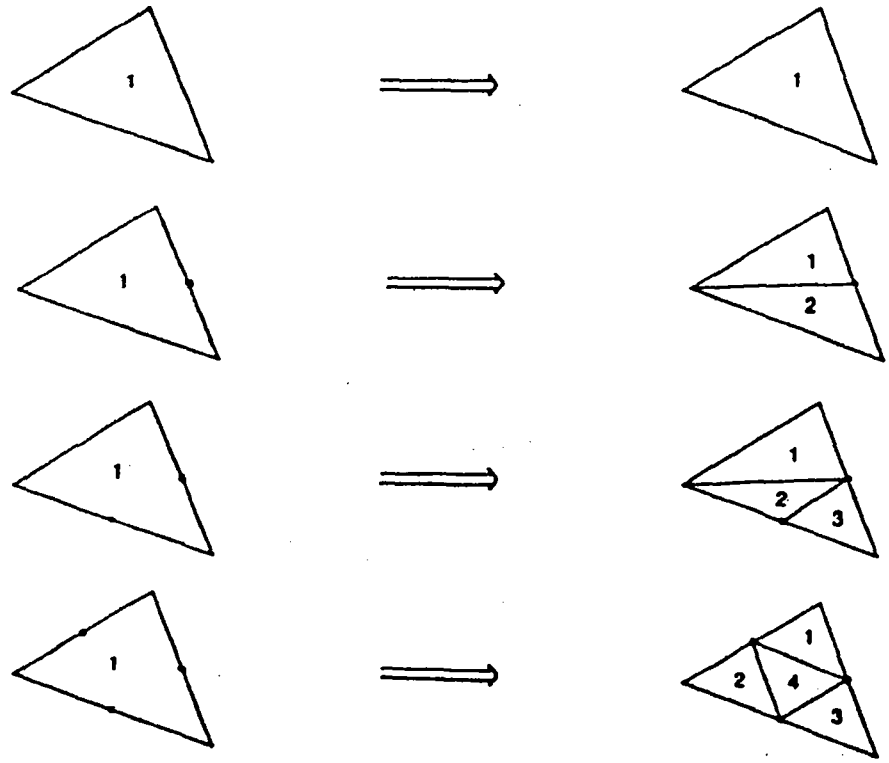


FIGURE 2 :
Triangle regeneration to obtain a conformal mesh

1.1.3. Criterion of enrichment

We consider a physical criterion highly advocated in steady Euler calculations. In these calculations, at the steady state, the entropy S should be purely convected, i.e. satisfying

$$\vec{U} \cdot \vec{\nabla} S = 0$$

where $\vec{U}$ the velocity of the flow, except in the vicinity of the shocks which may act as sources of entropy.

This observation leads us to define a new criterion to determine the region to be refined, i.e. on each triangle where the scalar product $\vec{U}.\vec{\nabla}S$ is greater than a constant C times the maximum of $\vec{U}.\vec{\nabla}S$ (taken over all the triangles). C is then a parameter controlling the level of refinement that we wish to perfom.

As a consequence, the regions of refinement will locate both the elements where spurious numerical entropy production occurs, and the physical entropy production on shocks.

1.2. *Application to Euler flow*

We present an application of this procedure when combined to the upwind scheme

descrided in these notes ; many experiments have also been done with a Galerkin scheme, we refer to Lohner et al [LOH2] and to [PAL3] and [ANG8].

1.2.1. A 2-D experiment

The purpose of the first experiment that we present is to show how smooth (oscillation-free) and low-entropy distributions can be obtained by applying (even strongly local) refinements : as a test case we consider a transonic flow around a NACA0012 airfoil with a freestream Mach number equal to $M_\infty = .85$, at zero angle of attack. The half-upwind scheme without limiter is used. We start with an initial mesh containing 600 nodes shown on Fig. 3. The first value of the grid refinement parameter C is 3% and we obtain an enri hed mesh with 1049 nodes presented on Fig. 4. Then, a second refinement is performed with the parameter C taken equal to 7 %, but this time, the grid refinement has been performed only in the front part to avoid the addition of too many points near the shock. By this procedure, a final mesh with 1615 nodes is obtained (Fig. 5). We can make the following comments : on one hand the entropy distributions presented on Fig. 6, 7, 8 maintains a rather smooth aspect at the leading edge, on the other hand, numerical entropy reduction is achieved in a monotone manner, with a ratio of about 3 for each refinement.

The final result gives an entropy level of the order of 10^{-3} almost everywhere before the shock.

It is interesting to compare the entropy distribution obtained with only one enrichment (1049 nodes, Fig. 7) with the distribution obtained with a full uniform splitting of the initial mesh which leads to a regular O-mesh with 2280 nodes (Fig. 9) ; the good agreement before the shock, demonstrates the efficiency of the local refinement approach.

Clearly the final resulting mesh is not the best available NACA0012 mesh ; our purpose is only to promote local self-adaptive refinement as a flexible and efficient method to generate physically adapted triangulations from arbitrary initial meshes.

Another remark can be done, in relation with the UPWIND-FEM scheme : the good convergence to zero of the entropy level seems to give a confirmation of the second-order accuracy of this non slope-limited scheme ; one disadvantage is the internal viscosity that need many nodes for the capture of the stagnation points ; one advantege is the smooth behavior in presence of strong mesh variation.

1.2.2. 3-D experiments

The 3-D case is handled in a similar manner; three ways to split a tetrahedron are considered :

- splitting in two tetrahedra (one face is divided in two triangles)

- splitting in four tetrahedra (one face in four triangles, three in two ones)

- splitting in eight (each face in four ones).

In order to keep a conformal mesh, the way to split a given tetrahedron cannot be decided independently on the neighboring ones: in a few sweeps over the mesh the number of sub-tetrahedra is increased till a globally conformal mesh is obtained.

An illustation of the power of the approach is presented in Figure 10 (adaptive mesh for a supersonic flow past a delta wing).

Since in 3-D the density of nodes is directly multiplied by a factor of eight, the method is both very powerful and not yet very easy to use. The above calculation shows in a typical manner that higher resolution is obtained in region that are rather far from the wing; this resolution may be unnecessary for the calculation of global coefficients (lift, drag,..).

2. MESH ADAPTION BY NODE MOVEMENT

In the previous section, we have applied a refinement method relying on mesh enrichment. We obtained good results in several two-dimensional and three-dimensional flow calculations. However these refinement algorithms are less satisfactory when we look for capturing thin layers such as shocks, slip lines, shear layers, because the refinement is not enough directional. In the three-dimensional case in particular, the very rapidly increasing number of mesh nodes resulting from enrichment represents a real handicap. In order to improve this approach, we may combine the enrichment method and a refinement method in which we move the mesh nodes while their number remains constant.

For recent Euler computations with this node movement method we refer to Nakahashi and Deiwert [NAK], Dwyer [DWY], Lohner [LOH2].

While in the enrichment method the solution remains consistent during the enrichment step due to the interpolation, in a moving node method, the interpolation can be costly in CPU time, and not completely vectorizable ; we present an algorithm combining some features of the ALE method [HIR2] and the spring method [GNO] to obtain an adaptive procedure in the context of unstructured finite element for the multi-dimensional cases. When the spring method is applied, the flow calculation may become inaccurate if a convenient interpolation of the flow between two successive meshes is not applied. Combining the spring method with the ALE method the flow is computed by a consistent time-dependent scheme : indeed, after

each remeshing, the flow calculation is corrected by an advection-type equation, related to the mesh velocity, that projects all the quantities on the new mesh.

2.1. *The spring method*

2.1.1. Mesh definition by spring equations

Recall that the meshes considered are Finite Element-type triangulations. The topology is given (and fixed) ; it is defined by the following informations : are known the indices of vertices for each triangle and the boundary logical pointers for all the vertices. Only deformations are then allowed. To move the mesh, we first assume that each mesh node i is connected to each adjacent mesh point j by a fictitious spring under the force F_{ij}. For all the nodes i of the mesh $\mathcal{M}_0$, for each j neigbor of i,

$$\overrightarrow{F_{ij}} = G\ K_{ij}\ \overrightarrow{ij} \tag{1}$$

in which G is an adjustable constant and K_{ij} the spring constant, both are positive numbers; K_{ij} is mode to depend on the flow properties (for adaption). The unknowns are the coordinates $(X(i),\ Y(i))$ of vertices ($i = 1, \cdots, N, N$ is the number of vertices). The resulting mesh is the solution of the equilibrium system, which is written, for each vertex i :

$$\sum_{j\ neighbor\ of\ i} \overrightarrow{F_{ij}} = 0\ . \tag{2}$$

2.1.2. Flow criterion

How can K_{ij} be best chosen ?. Two kinds of adaptive criteria are generally used :

(I) the mesh can be adapted from an estimation of the local approximation error, the objective being to reduce it.

(II) the mesh can be adapted in order to have a better representation of some sensor S related to the flow.

Since we are interested in directional adaptation, a criterion belonging to the second kind (II) is more straightforward to construct and analyse. We plan to discuss the first option (and other options) in a future publication. Since shock capture is to be applied, we may choose S as the Mach number M. It is assumed to be numerically defined on each vertex i. More precisely, in a first study, we put , for two adjacent nodes i, j :

$$K_{ij} = |S(i) - S(j)| \tag{3}$$

which approximates the directional variation of the sensor S.
Also a smoother value of the spring constant K_{ij} can be introduced to take into account values at each neighbor of the nodes i and j : we introduce the following derivatives :

$$(4) \qquad \begin{cases} \dfrac{\partial S}{\partial s}(i,j) = \dfrac{S_{(i)} - S_{(i)}}{\|ij\|} \\ \dfrac{\partial S}{\partial s}(i) = \overrightarrow{\nabla S}_{(i)} \cdot \dfrac{\overrightarrow{ij}}{\|ij\|} \\ \dfrac{\partial S}{\partial s}(j) = \overrightarrow{\nabla S}_{(j)} \cdot \dfrac{\overrightarrow{ij}}{\|ij\|} \end{cases}$$

with

$$(5) \qquad \overrightarrow{\nabla S}(i) = \begin{cases} \dfrac{\int\int_{Supp\ \phi_i} \frac{\partial S}{\partial x}\phi_i \, dx \, du}{\int\int_{Supp\ \phi_i} \phi_i \, dx \, dy} \\ \dfrac{\int\int_{Supp\ \phi_i} \frac{\partial S}{\partial y}\phi_i \, dx \, dy}{\int\int_{Supp\ \phi_i} \phi_i \, dx \, dy} \end{cases}$$

ϕ_i is the P_1 basis function corresponding to the vertex or node i, $Supp\ \phi_i$ is its support. If we denote :

$$(6) \qquad \begin{cases} d_i = \dfrac{\partial S}{\partial s}(i).\|ij\| \\ d_j = \dfrac{\partial S}{\partial s}(j).\|ij\| \\ d_2 = |S(i) - S(j)| \end{cases}$$

and introduce

$$(7) \qquad \begin{cases} d_1 = 2d_i - d_2 \\ d_3 = 2d_j - d_2 \end{cases}$$

Then we choose K_{ij} as a barycenter of the variations d_1, d_2, d_3 namely :

$$(8) \qquad K_{ij} = \frac{1}{3}(|d_1| + |d_2| + |d_3|)$$

2.1.3. Solution algorithm for the spring equations

The coordinates $X(i)$, $Y(i)$ of the vertices derive from equations (1),(2). They are then the solution of a linear system :

$$(9) \qquad AX = B_1, \ AY = B_2$$

with

$$(10)\qquad \begin{cases} A_{ij} = -G.K_{ij} \quad for\ i \neq j,\ j\ neighbor\ of\ i \\ A_{ii} = \sum_{j\ neighbor\ of\ i} G.K_{ij} \end{cases}$$

A is a M-Matrix in a weak sense, that is, it satisfies the following properties :

$$(11)\qquad A_{ii} \geq 0\ ;\ A_{ij} \leq 0\ for\ i \neq j,\ \sum_j A_{ij} \geq 0\ \forall_i$$

More precisely, for an internal node with all neighbors internal, (A is weakly positive) :

$$(12)\qquad \sum_{j\ neighbor\ of\ i} A_{ij} = 0$$

For the nodes i which have boundary nodes neighbors (A is positive) :

$$(13)\qquad \sum_{j\ neighbor\ of\ i} A_{ij} > 0\ \ if\ \ G\,K_{ij} \neq 0$$

In this case, provided that the A_{ij} in Equations (10) are not all vanishing, the equation gives a right hand side which corresponds to boundary vertices that we assume to be either fixed or moving along the boundary.

Well posedness and notion of reference mesh

Let us assume that the flow solution is uniform in some region of the computational domain, then the K_{ij} are identically vanishing in that region, so that the node system (9) is not well-posed. Then we have to add some device in order to well define the mesh in such regions. In fact this means that the algorithm must be able to construct an admissible mesh in the particular case of a globally uniform flow ; let us call "reference mesh" the resulting mesh in that case : note in particular, that the present algorithm can be used as a mesh generation method. We have introduced two new options for constructing an admissible reference mesh.

Residual strength

The K_{ij} (eq (8)) are replaced by

$$(14)\qquad \bar{K}_{ij} = K_{ij} + C_0,\ C_0 > 0$$

C_0 is a constant parameter.
The corresponding reference mesh is defined by the equilibrium of springs with

equal strength. Such meshes tend to attain some "optimality" with respect of angles (equilaterality) and (or) repartition (location of nodes).

Connection to initial mesh

It is possible to enforce diagonal dominance in the matrix A when we wish to adapt the mesh while maintaining it close to the initial mesh. In this purpose, additional terms are easily introduced by modifying the equilibrium Equation (2) as follows : for each node i,

$$(15) \qquad \Big(\sum_{j \text{ neighbor of } i} F_{ij} \Big) + F_i^{in} = 0$$

with

$$(16) \qquad F_i^{in} = (F_i^{in,X}, F_i^{in,Y}),\ F_i^{in,Z} = C_1(Z(i) - Z_0(i)),\ Z = X \text{ or } Y$$

where $(X_0(i),\ Y_0(i))$ are the coordinates of the initial mesh ; C_1 is a constant parameter to be adequately chosen.
In [X] we are given some examples of reference meshes.

Maximum principle

• From the properties of the matrix A (see Eqs (11) and (12)), we deduce that if the computational domain is convex, then all the vertices of the resulting mesh will be inside the domain. However this does not guarantee that the triangles will not overlap, even in the convex case. Therefore some control and monitoring will be necessary to prevent overlapping.

• Using the equilibrium system defined by Equation (13), with spring stiffness introduced in Equation (14) to determine the new mesh, we obtain a linear system whose matrix A is strictly diagonally dominant. We are interested in performing a high number of remeshing : to limit the C.P.U. time, we solve the mesh system by a few Jacobi iterations here. This choice is explained as follows : in the Gauss-Seidel itetation, we note the first node is moved only depending on the old position of its neighbors while the last node comes from the updated values of all the nodes. Thus partial convergence of the iteration process will introduce dissymmetries in the new mesh which, in particular becomes inconvenient in the unsteady case. With Jacobi iterations the system (9) converges more slowly than with Gauss-Seidel iterations but the non converged results do not depend on the node numbering, and symmetry properties are preserved (furthermore the vectorization is easier but it has not been attempted in this work).

2.2. *Limitation of the displacement of each node*

An admissible mesh is not surely attained with the spring method : when we try to determine thin layers, the mesh may overlap, particularly in the case where

the refined zone becomes very fine for capturing highly curved layers. For treating this case, we introduce the following conditions which do not yet systematically guarantee the mesh admissibility, but are not very costly to enforce : for each node i we construct the following safety area : the sphere with center i and a radius equal to k times the smallest height h_i drawn from i in the corresponding cell C_i, k is a real positive coefficient which has to be smaller that 0.5). Finally, we lower the displacement of all the nodes so that each node remains in its safety area as follows:

$$\|\overrightarrow{\Delta M_i'}\| = k\|\overrightarrow{\Delta M_i}\| \ , \tag{17}$$

with k given such that for all i

$$k\|\overrightarrow{\Delta M_i}\| \leq 0.5\ h_j \tag{18}$$

2.3. *ALE mesh-to-mesh conservative transfer*

Our goal is to obtain conservative computation of the flow corresponding to the mesh obtained after the spring step. We realize this by the mean of a discretized equation in which we have to take into account two phenomena :
- the mesh movement, represented by a n de velocity.
- the geometrical change of the mesh, appearing in the areas of the cells.

Let us call $\mathcal{M}^{n-1}(W^n)$, the mesh (the flow computation in $\mathcal{M}^{n-1}$) before the spring-limited step and $\mathcal{M}(W^n)$, the final one (the flow computation in $\mathcal{M}^n$), $\bar{W}^n$ is the contribution to the flow, due to the mesh movement :

$$(\mathcal{M}^n,\ \bar{W}^n) \ = \ P_n\ (\mathcal{M}^{n-1},\ W^{n-1}) \tag{19}$$

in which P_n is the transfer operator from the $(n-1)^{n-1st}$ mesh to the n^{th} mesh. The application of P_n is equivalent to solving the advection equation with a discretization domain defined by $\mathcal{M}_n$:

$$\bar{W}_t \ - \ \vec{V}^{\,n-1} . \nabla \bar{W} \ = \ 0 \tag{20}$$

$\vec{V}^{\,n-1}$ is the velocity of the mesh $\mathcal{M}^{n-1}$. Actually we do not use a discretization of the equation (20) : as we have to preserve the conservativity, we are solving an equation of the type :

$$(area\ (C_i)\ \bar{W}_i)_t \ - \ div(\vec{V}_i\ \bar{W}_i) \ = \ 0 \tag{21}$$

C_i is the cell around node i. $\vec{V}_i$ is the mesh-velocity of the node i. The equation is written for all the nodes i in the mesh $\mathcal{M}^n$. The introduction of the cell area

corresponds to the Jacobian of the mapping from $\mathcal{M}^{n-1}$ to $\mathcal{M}^n$. We can explicit equation (21) by :

$$area(C_i^n)\ \bar{W}_i^n \ - \ area(C_i^{n-1})\ W_i^{n-1} \ = \ \Phi_i^{n-1} \tag{22}$$

with

$$\Phi_i^{n-1} \ = \ \sum_{j \ neighbor \ of \ i} \int_{\partial C_i^{n-1} \cap \partial C_j^{n-1}} W_{aj}^{n-1} . D_i^{n-1} . \dot{n} d\sigma \tag{23}$$

$$D_i^{n-1} \ = \ \vec{V}_i^{\,n-1} \Delta t \tag{24}$$

- C_i^n is the cell around the node i in the mesh $\mathcal{M}^n$. ∂C_i^n denotes the boundary of cell C_i^n.

- W_i^n is the value of W, at the node i in the mesh $\mathcal{M}^n$.

- $\vec{V}_i^{\,n-1}$ is the velocity of the node i in the mesh $\mathcal{M}^{n-1}$, D_i^{n-1} corresponds to the displacement

- $\dot{n}$ is the outward normal to $\partial C_i^{n-1} \cap \partial C_j^{n-1}$ of the node i in the mesh $\mathcal{M}^{n-1}$.

The value of W_{aj}^{n-1} in the equation (23) is determined by the solving(21) with one of the following two upwind schemes.

In the first-order accurate case, we put :

$$W_{aj}^{n-1} \ = \ \begin{cases} W_i^{n-1} \ \ if \ \ \int_{\partial C_i^{n-1} \cap \partial C_j^{n-1}} D_i^{n-1} \ n \ d\sigma \ > \ 0 \\ W_j^{n-1} \ \ if \ \ \int_{\partial C_i^{n-1} \cap \partial c_j^{n-1}} D_i^{n-1} \ n \ d\sigma \ < \ 0 \end{cases} \tag{25}$$

In the second-order accurate case, W_{ij}^{n-1} and W_{ji}^{n-1} are defined at the mid point I_{ij}^{n-1} of $\partial C_i^{n-1} \cap \partial C_j^{n-1}$ with the MUSCL-FEM interpolation introduced in Chapter I derived from the W_i^{n-1}, W_j^{n-1}, etc. Moreover we use limiters with the physical variable $\hat{W}$ (introduced in Chapter I). After computing of $\bar{W}_n$ (equation (20)), we first reactualize the metrics, i.e. the cell areas, the gradient of the basis functions, the coefficients corresponding to the discretization above each side of the triangulation. We finally obtain the flow solution W_i^n on the new mesh from the formula :

$$W_i^n \ = \ (\ area(C_i^{n-1}) . W_i^{n-1} \ + \ area(C_i^n) \bar{W}_i^n \ ) \ / \ area(C_i^n) \tag{26}$$

2.4. *Organization of the algorithm*

The procedure to couple the flow calculation and the mesh adaption is given by the following algorithm :

0 - initialization of the mesh (the mesh index, imesh, is equal to 0) and of the Euler system.

1 - Flow calculation.

2 - Mesh index updating : imesh = imesh + 1.

3 - First calculation of the displacement of mesh nodes by springs.

4 - Limitation of each displacement obtained in Step 3.

5 - Final conservative calculation of the mesh and of the metrics.

6 - Go to Step 1.

We have to rely the different steps of this algorithm by taking into account two requirements :

- the CPU time should be limited

- the final pair (mesh, flow) should be well convergent.

The first point is attained from the following ways :
- several iterations in the flow calculation (Step 1)
- a few iterations in solving the spring system by the Jacobi iteration (typically, five iterations (Step 3)).

The second point is obtained by starting from an initial flow well-converged on the initial mesh and by a performing a sufficiently large number N of remeshings. At each new mesh, we compute the mesh residual R_M^n :

$$R_M^n = \|X_i^n - X_i^{n-1}\| \tag{27}$$

X_i^n is the limited position of the node i (end of Step 4) ; X_i^{n-1} is the previous position. The L_2 norm is used. The well-convergence of the algorithm is measured by the distance from R_M^n to zero. Near convergence, the mesh does not change very much, and the global algorithm proceeds essentially for the full convergence of the

flow calculation.

2.5. *Numerical illustrations*

The calculations presented here are done with an explicit pseudo-unsteady solver equivalent to the first phase of the implicit scheme described in Chapter II.

2.5.1. The shock tube test

We give here a classical example of unsteady flow calculation in two cases : in Fig.5, the results obtained without adaption, in Fig.6 the results obtained with mesh motion, using the algorithm defined in Section 5. The initialization of the flow is uniform from the boundary condition. The flow calculation is realized by using the second-order accurate scheme described in 1.2.2. with Osher's splitting (see 1.2.1). In the adaption case the number of solver iterations is equal to one. The time-stepping is uniform and the C.F.L. is equal to 0.5. In Step 3 (adaption case), the sensor is the Mach number (see Section 2.2) and we choose, parameters as follows :

$$G = 5000 \ (Equation \ (1))$$

$$C_0 = 1 \ (Equation \ (14))$$

$$C_1 = 0 \ (Equation \ (16))$$

The number of Jacobi iterations is equal to 5. In the adaption case, the time $t = 0.16$ is attained after 629 remeshings. We observe in the Fig.11, the final mesh is well refined in the shock zones. This calculation is too much costly, but demonstrates the consistency of the ALE formulation.

2.5.2. External supersonic flow around a cylinder

The freestream Mach number is equal to 8. The C.F.L. is equal to 0.5. The values of the parameters are the same as in the above case, except for the number of iterations in Step 1, which is equal to 50. In Step 5 the flow projection is performed by the first-order accurate scheme explained in Section 4. The final mesh is obtained after 25 remeshings. Clearly the final mesh and the isomach contours put in evidence a good capture of the bow shock (Fig. 12 - 13).

Another illustration has been obtained by performing first two successive enrichment phases before moving the nodes. This enables to evaluate the robustness of the moving method when it is applied to an unstructured mesh. As flow solver, a first-order accurate version is used. At each step of the experiment we show the mesh and the corresponding Mach contours : Step one, we start from the structured radial mesh with 1088 nodes of Fig. 12. Step two, we perform one enrichement in regions where entropy is generated. Step three, a second mesh enrichment is performed, to

better capture the rear details. Now. we see that it is not yet easy to recognize what happens at the rear side of the cylinder. Step four, the mesh movement adaption is applied. It appears that the movement is mainly restricted in front, to the bow shock, and in the rear, to the shocks and the slip lines (Fig. 14). The resulting Mach contours (Fig. 15) show a much thinner bow shock, and the angle between the rear shock and the separation slip line is more clear.

2.5.3. External supersonic flow around a NACA0012 (Figures 9,10,11,12)

The freestream Mach Number is equal to 2., the angle of attack is equal to 10 degrees; the CFL number is equal to 0.6 ; the scheme used for the flow calculation is the second-accurate one. Before the mesh adaption by the presented method, we perform two successive enrichments. The final mesh is obtained after 30 remeshings ; Fig. 16 and 17 give an idea of the improvement obtained.

2.5.4. Interacting jets

A last illustration is a calculation of the interaction of two supersonic jets in a partly open chamber. The Mach number at the entry of the jets is 1.5 . The main new feature in this test is that slip lines are the main detail of the flow. Two enrichment phases (based on the Mach number gradient) have been performed before the node displacement. Most part of the slip line are captured with a reduced numerical thickness, also the aspect ratio of the corresponding elements is not very high (Fig. 18).

2.6. *Conclusion*

The adaptive procedure requires a small fraction of the computational cost - the advantage of the ALE transfer is that the shocks are well handled during the flow time evolution : this allows to solve adequately the flow and the shocks in steady cases by unsteady approaches or in unsteady cases. Note that the method for moving the grid requires a sufficiently large number of almost well distributed points: this leads us to perform an enrichment step before beginning the mesh movement phase. In that case, the use of movement is particularly efficient as a mesh smoother.

3. NOTES

The first section is taken from [PAL3] except 1.2.2. which comes from [HS these] , the extended from a work of method was extended from a work of V. Selmin [VSSD]. The second section is essentially derived from a paper by B. Palmerio [Palmerio]. Another way to adapt the mesh is to re-construct it globally ; see [ZIE2].

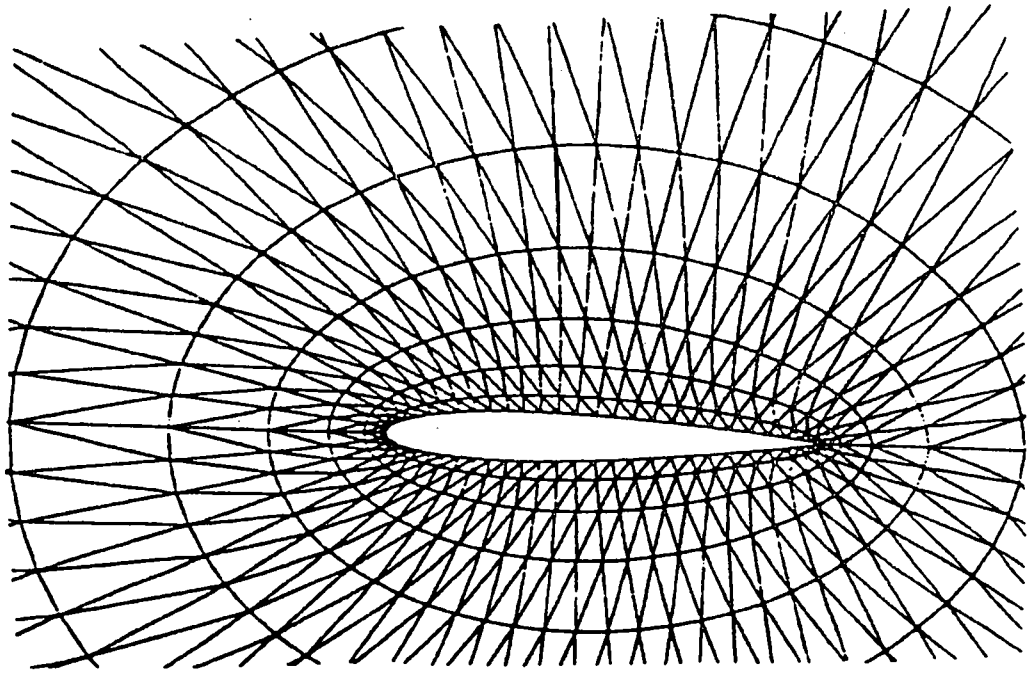


FIGURE 3 : Initial mesh, 600 nodes

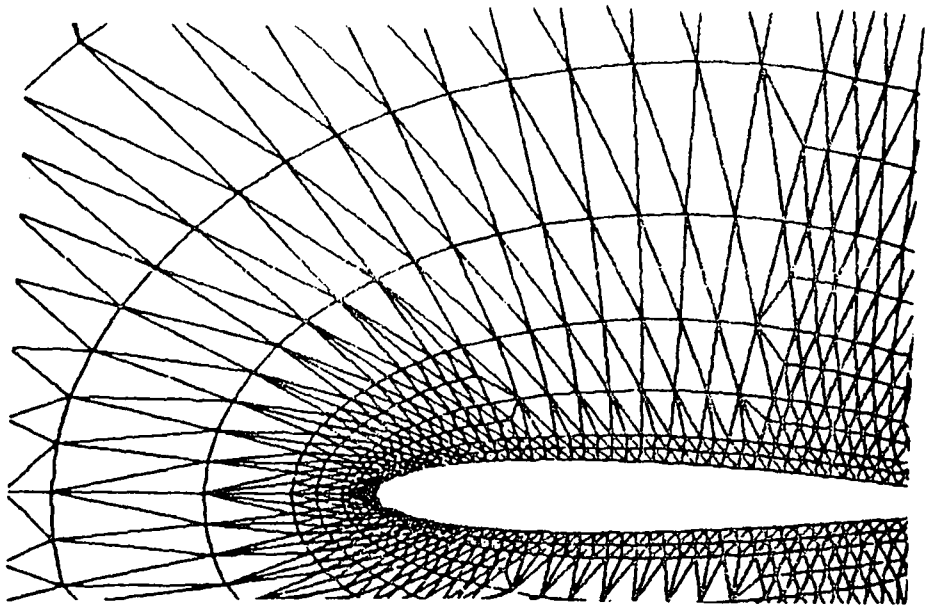


FIGURE 4 : Mesh refined once, 1049 nodes

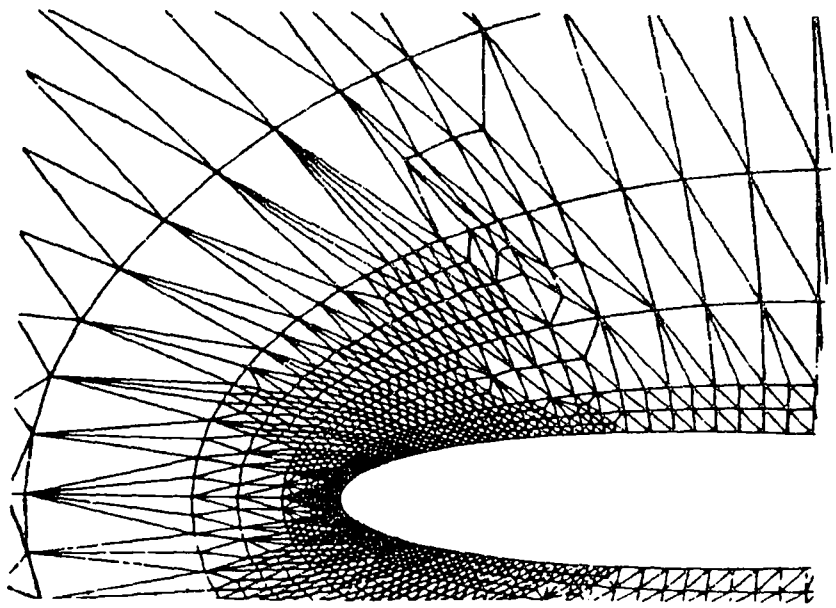
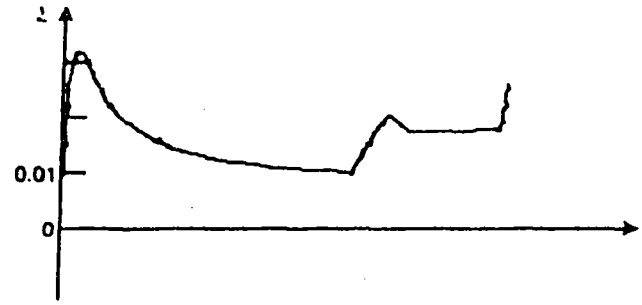


FIGURE 5 : Mesh refined twice, 1615 nodes

FIGURE 6

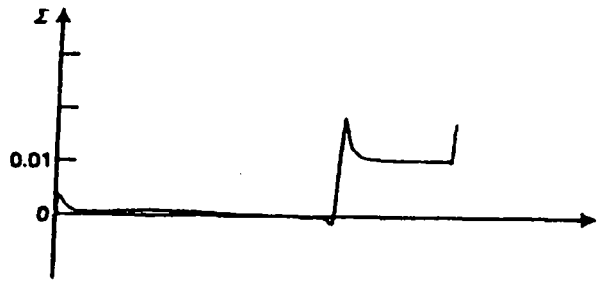


Mesh A, 600 nodes

FIGURE 7

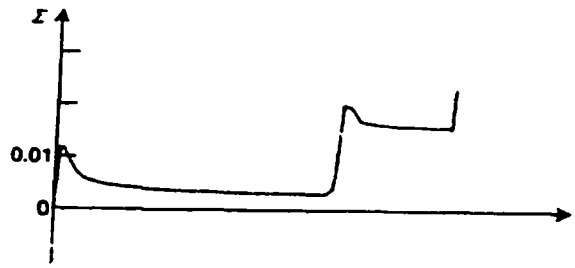
Mesh B, 1049 nodes

FIGURE 8



Mesh C, 1615 nodes

FIGURE 9



Mesh D, 2280 nodes

FIGURES 6 to 9
Comparison of entropy distribution along the airfoil ; mesh D is obtained by global division of the initial mesh A.

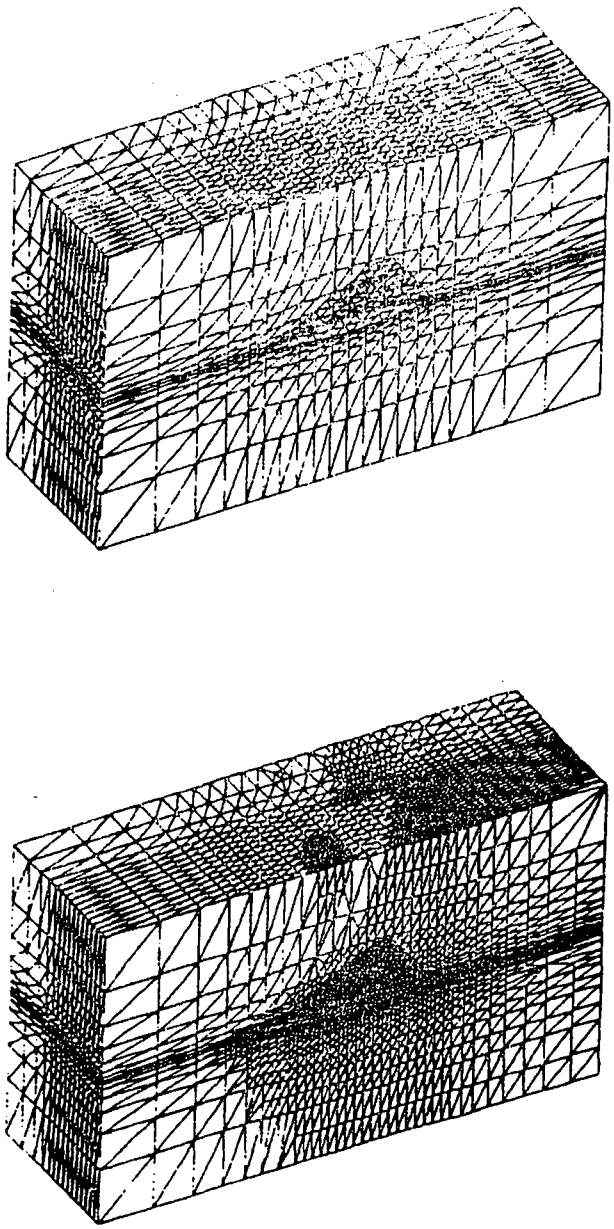
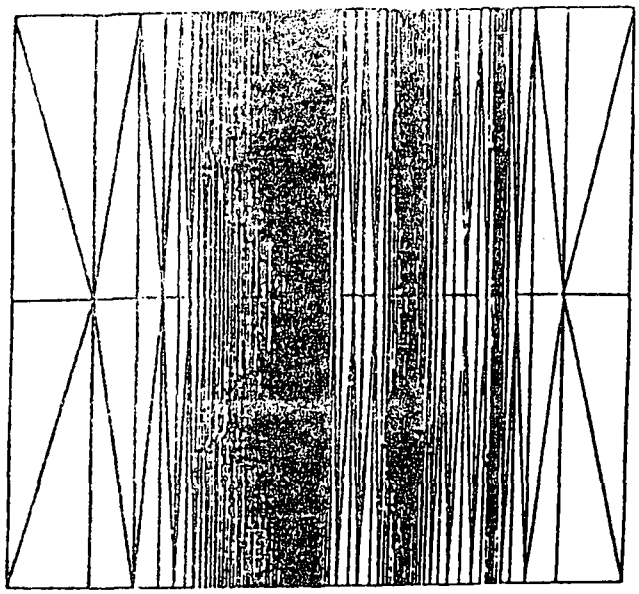
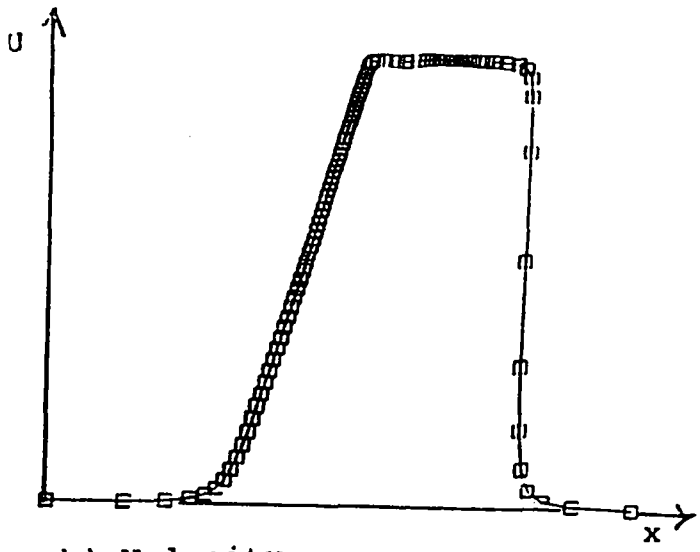


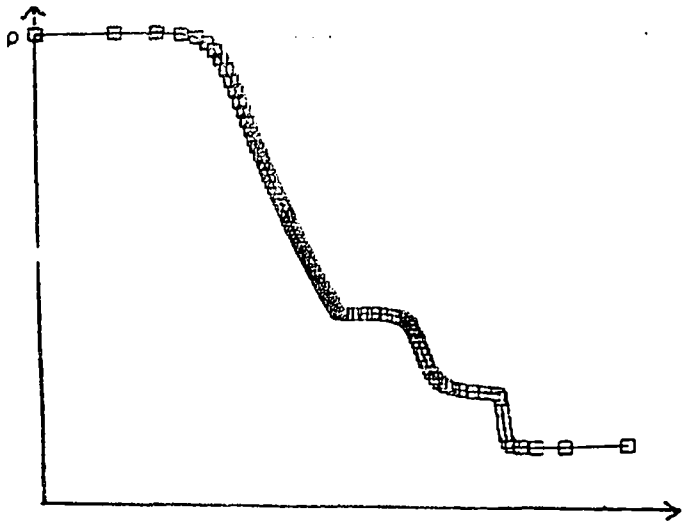
FIGURE 10 :
An illustration of the 3-D mesh enrichment algorithm ; the coarse mesh contains 5089 nodes ; the fine one contains 22160 nodes



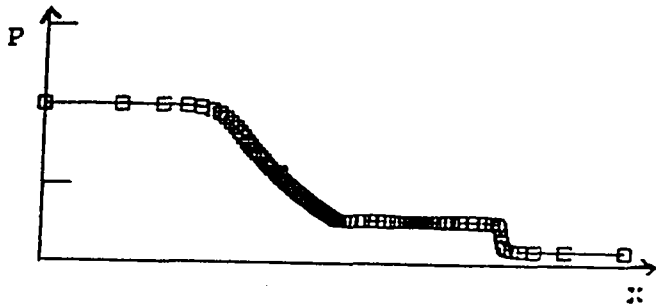
a) Final mesh (101x3 points)



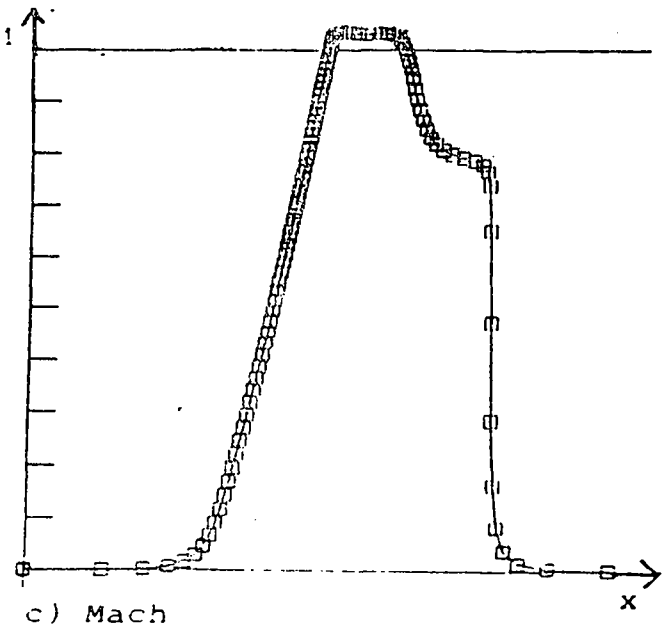
b) Velocity



c) Density

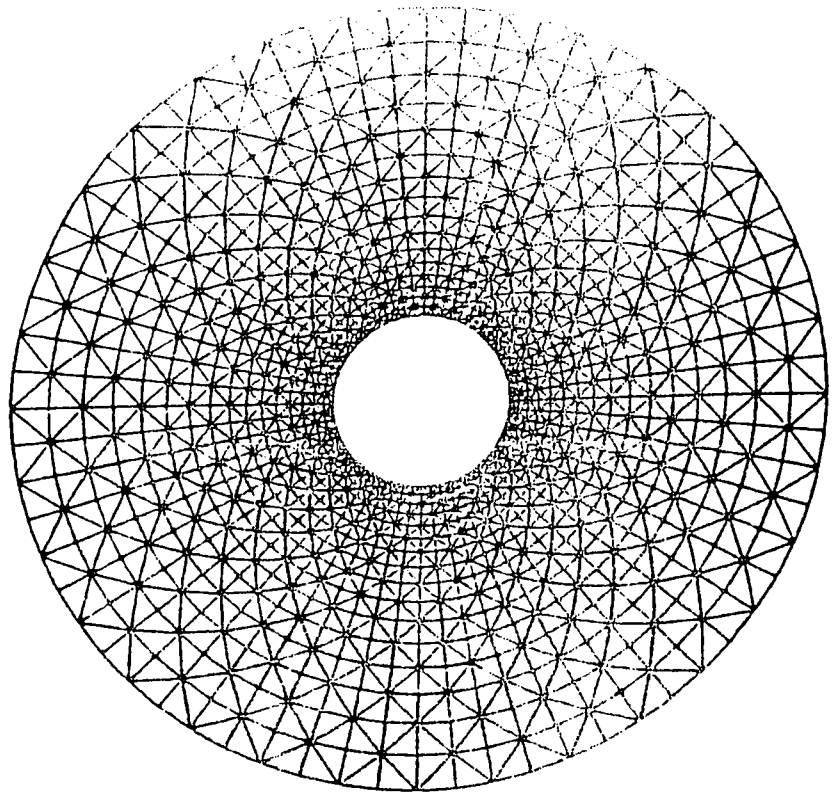


d) Pressure

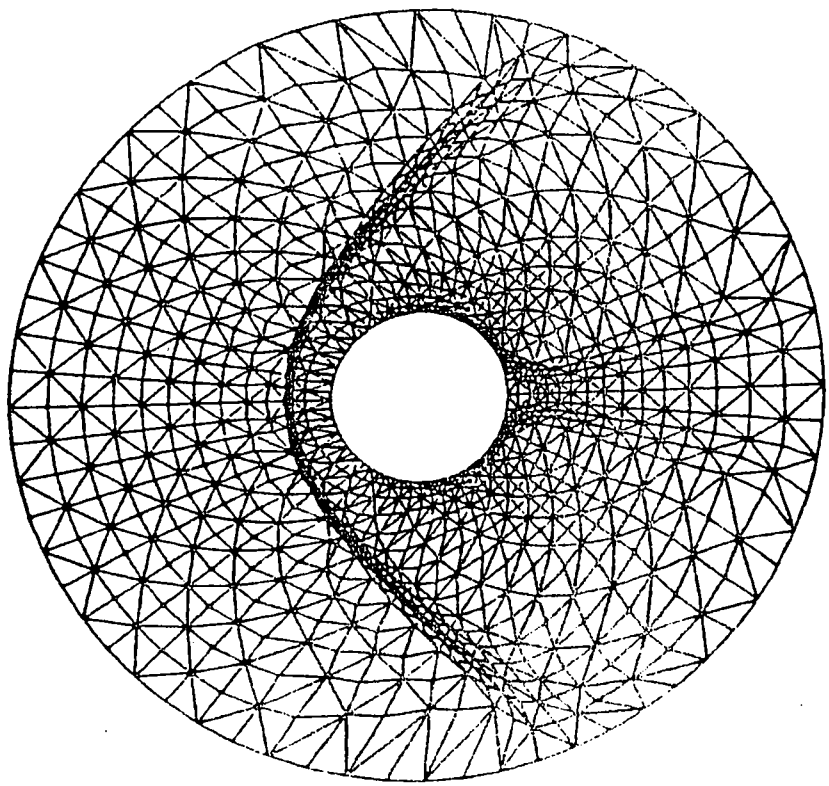


e) Mach

Figure 11: The shock tube test (Osher second-order accurate scheme, t = 0.16).

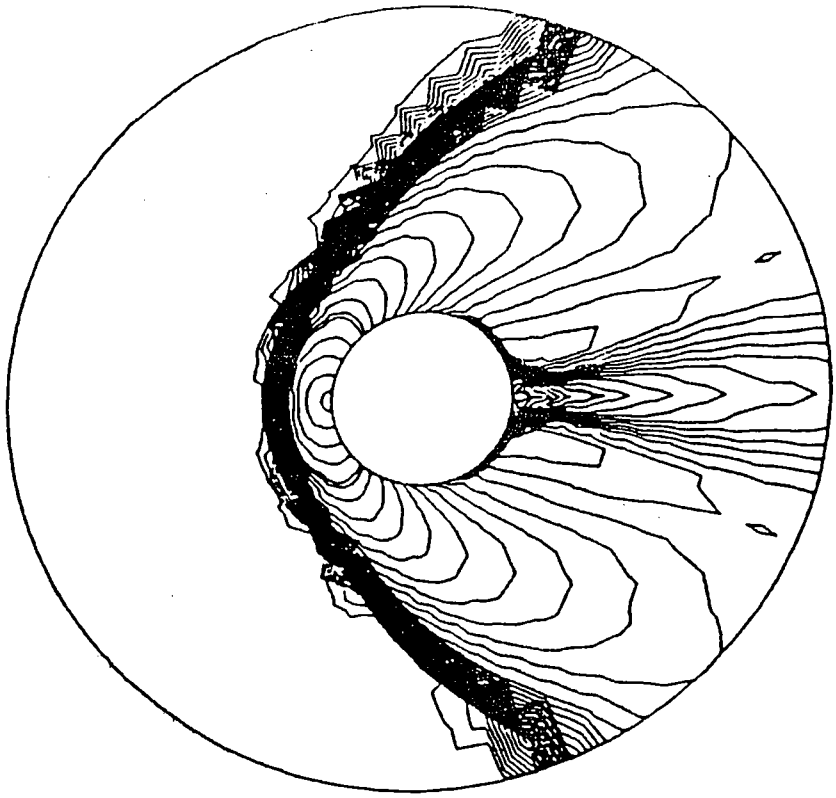


a) Initial mesh (1088 points)

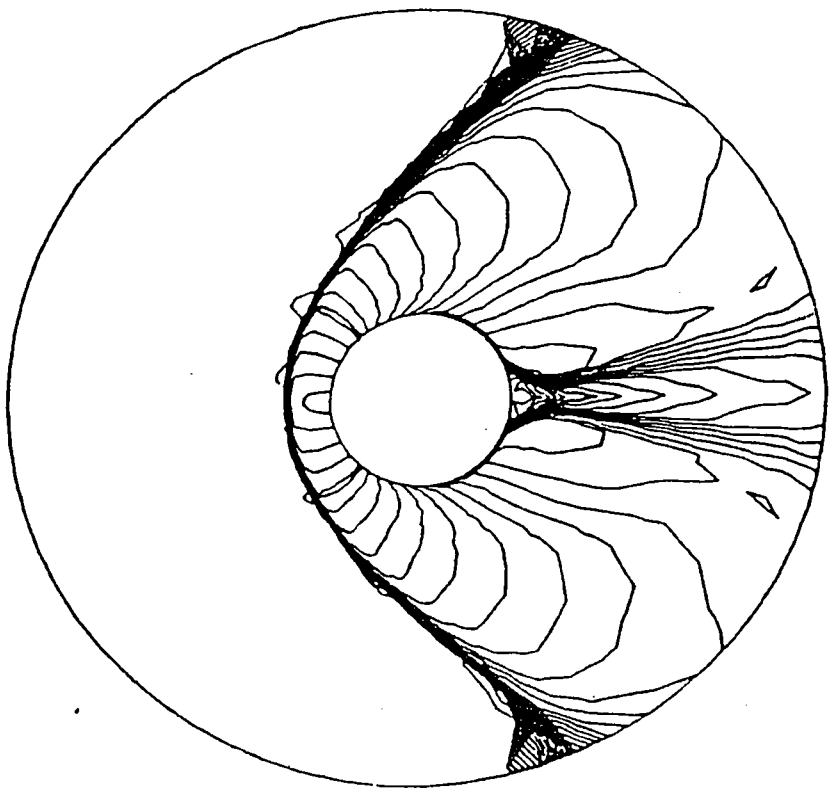


b) Final mesh

Figure 12 : External flow around a cylinder with a Mach at infinity equal to 8., mesh movement with ALE



a) Without adaption



b) with mesh moving

Figure 13 : Isomach contours corresponding to Figure 12

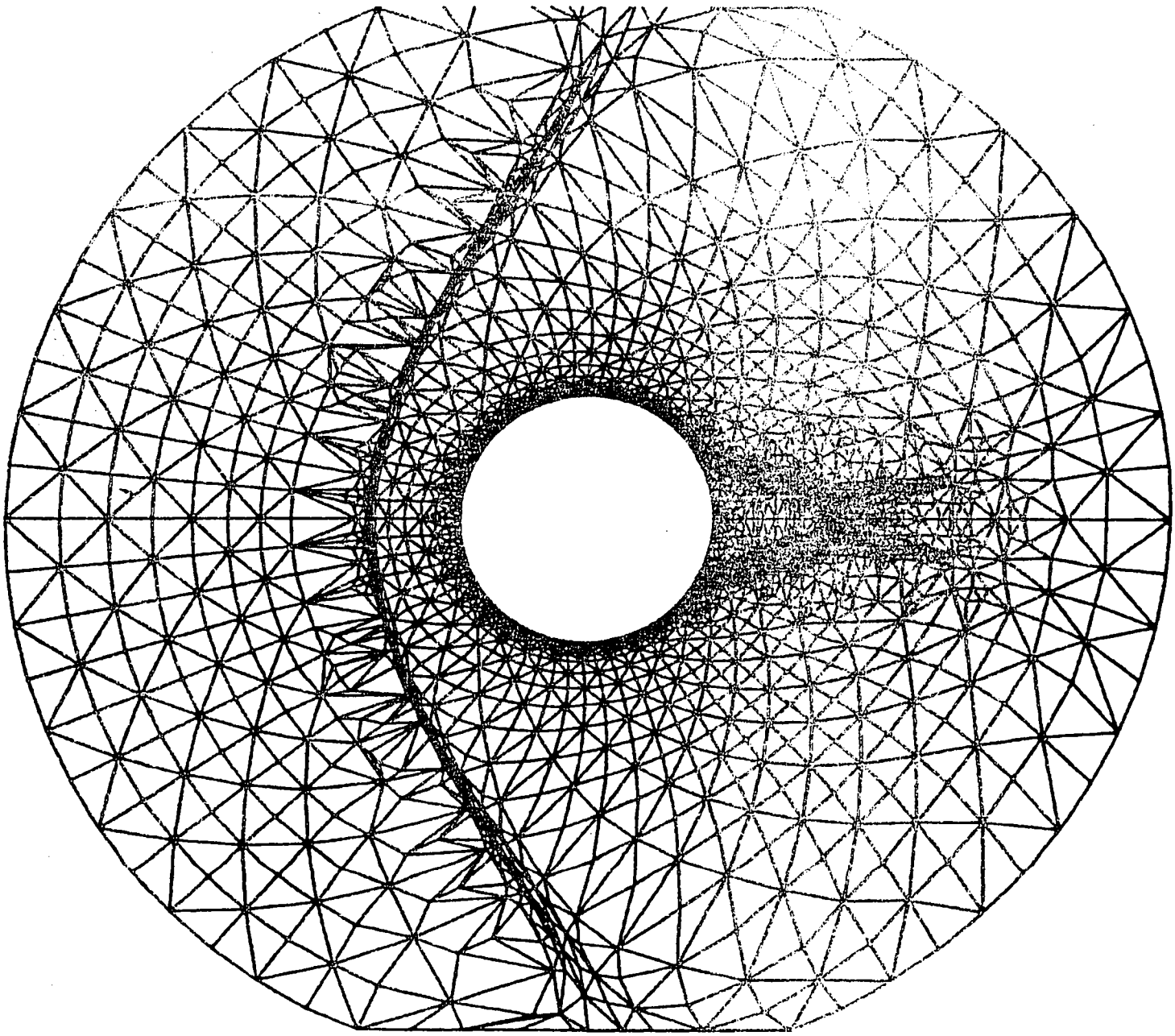


FIGURE 14 : Final mesh (2448 nodes) after two enrichment phases and the movement one.

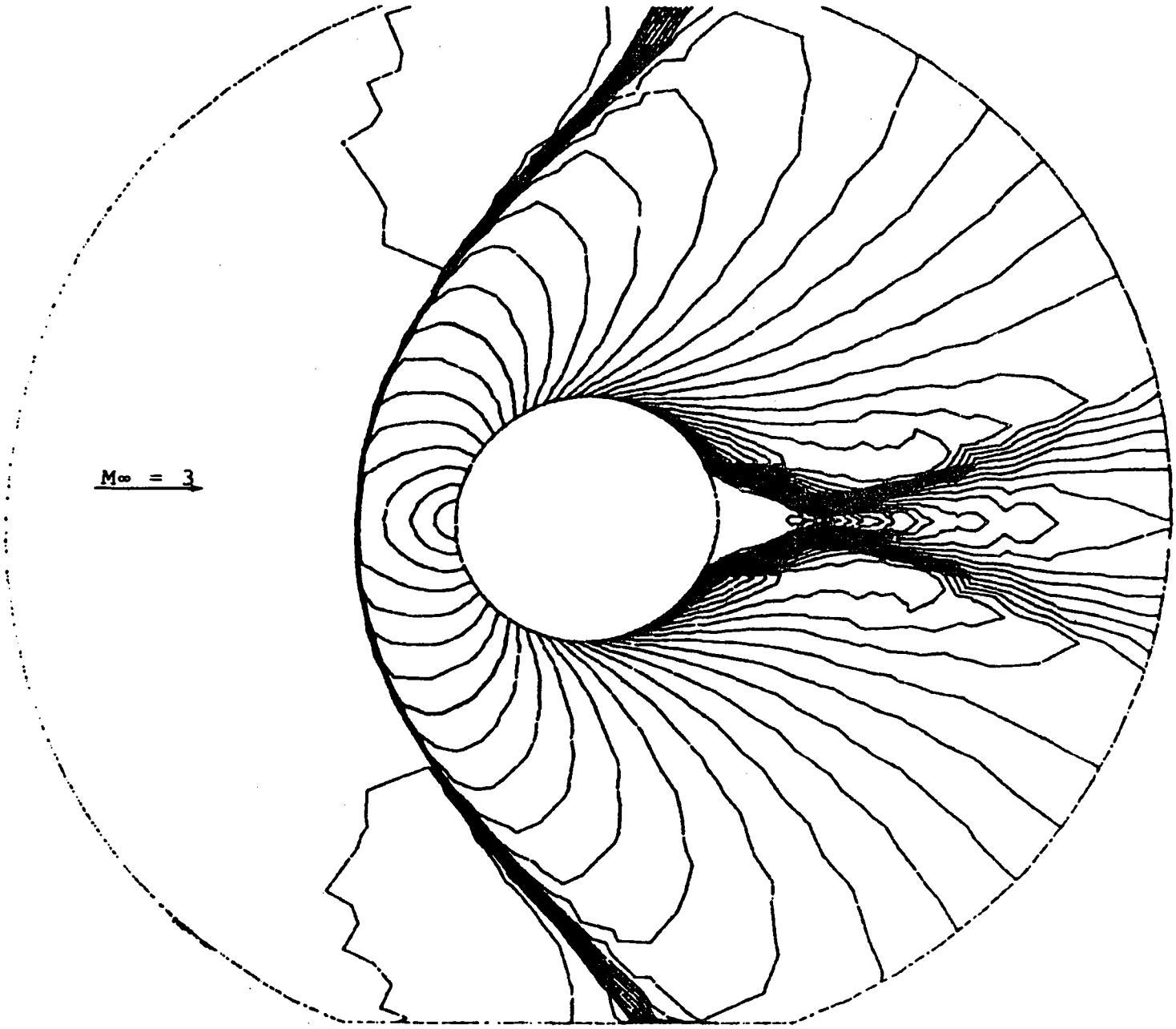
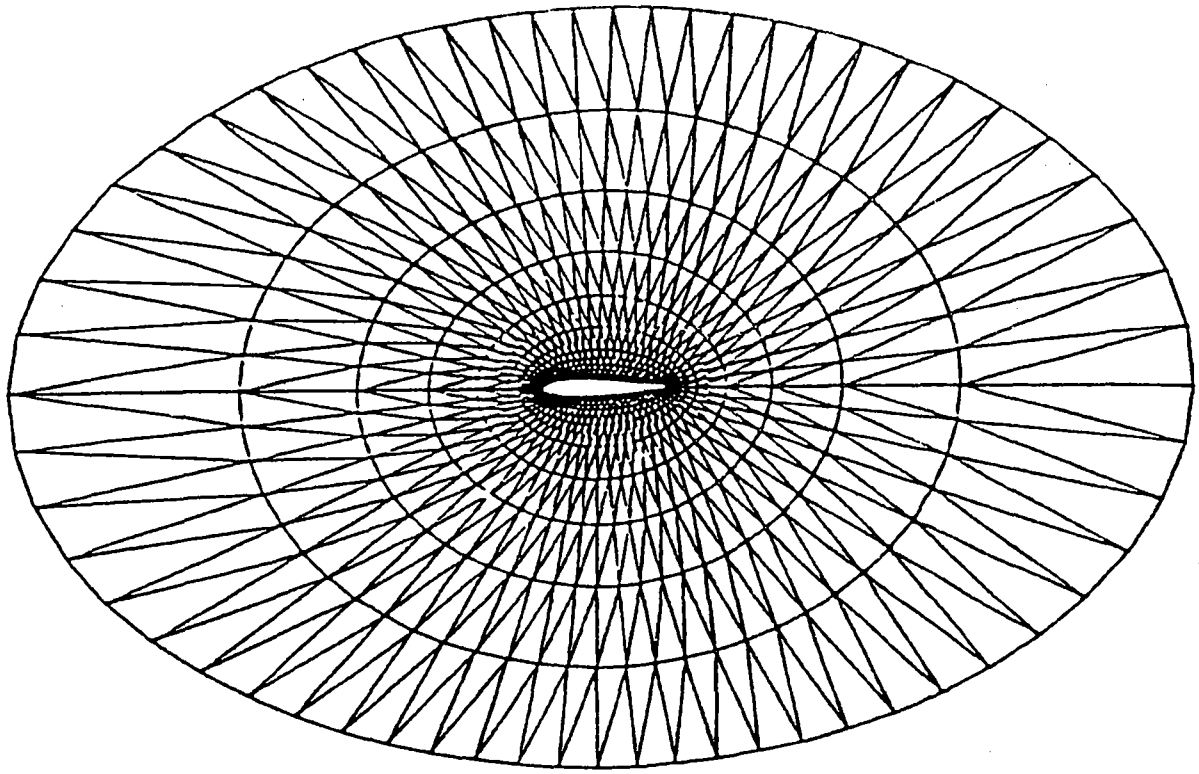
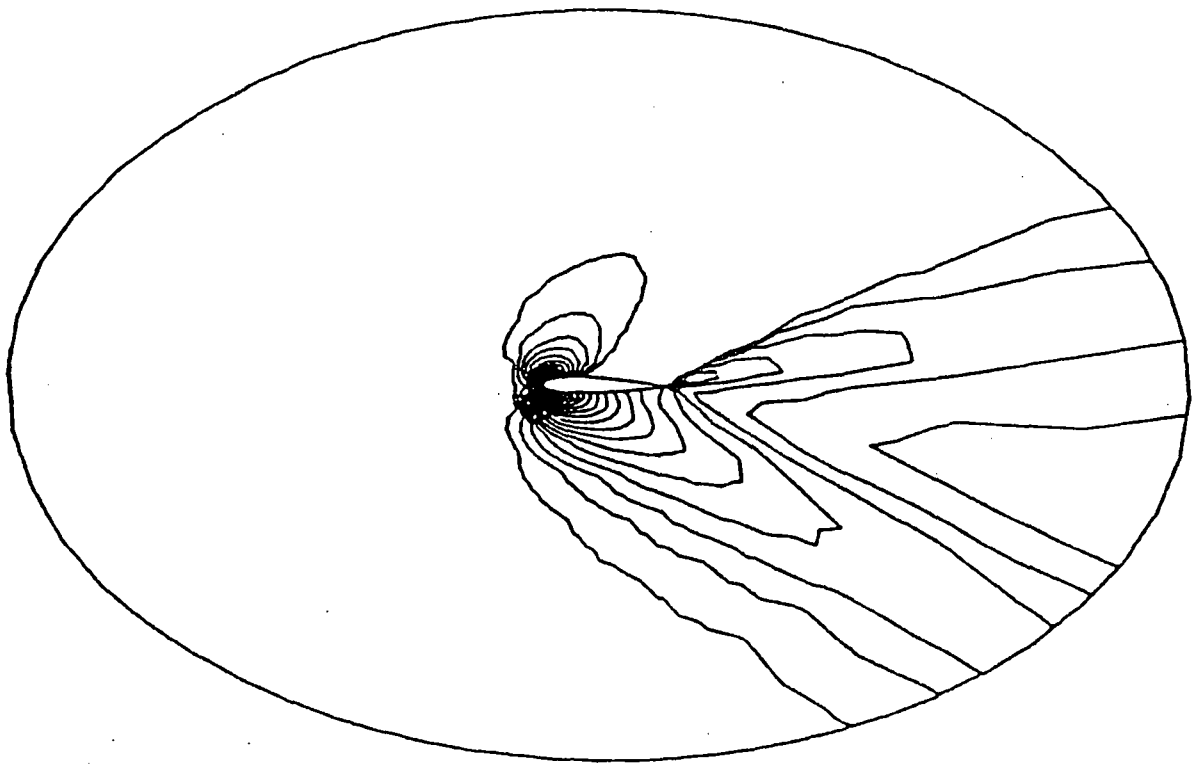


Figure 15 : Mach number contours for the mesh in Fig. 14.

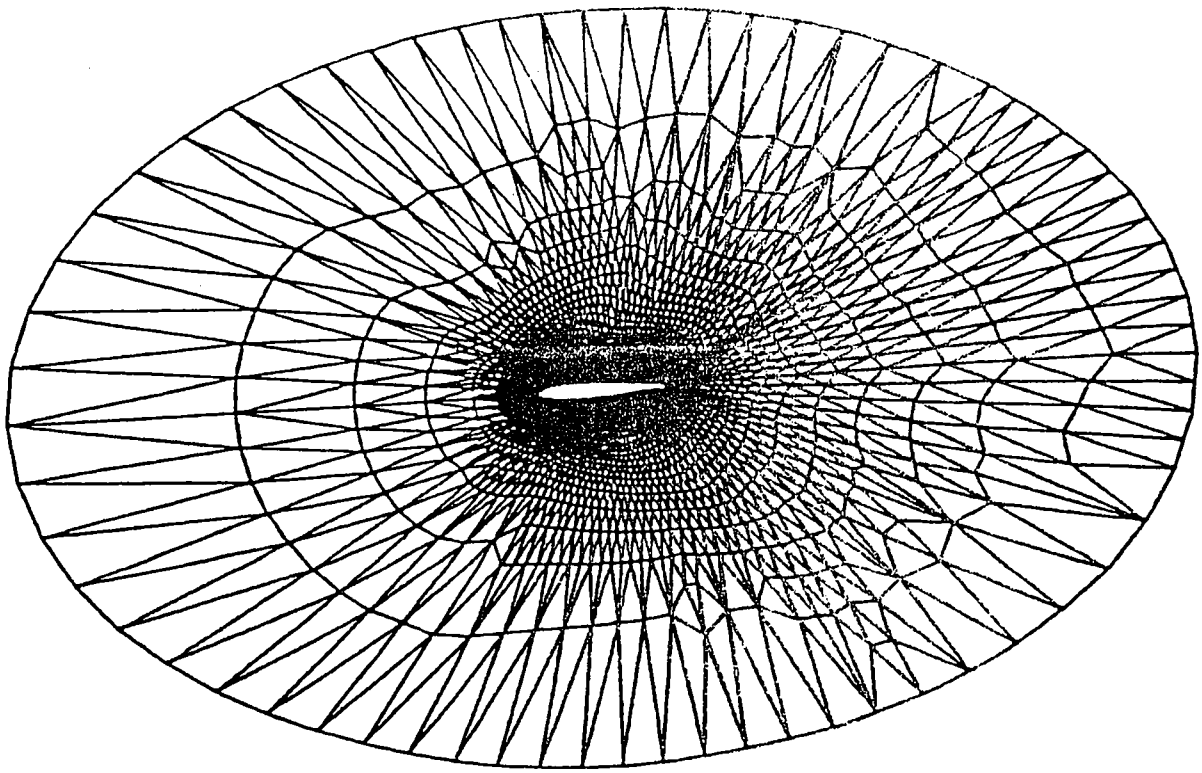


a) Initial mesh (600 points).

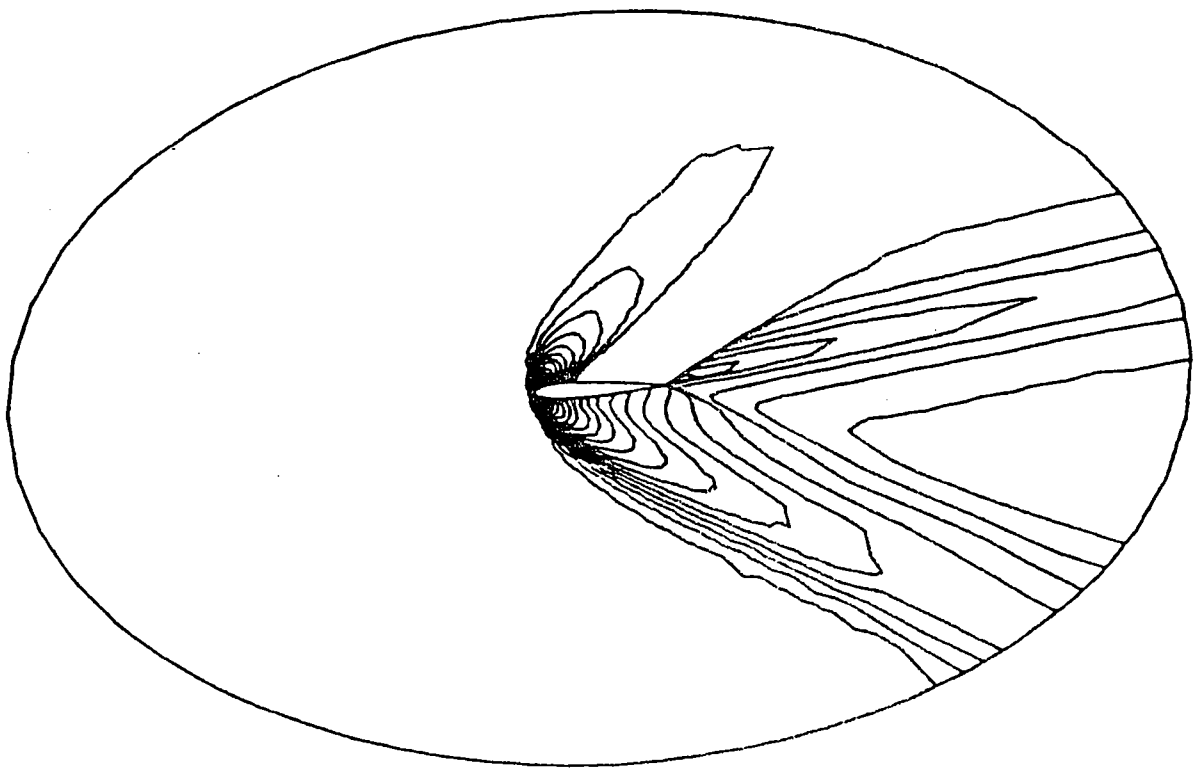


b) Isomach contours

Figure 16: External flow around a NACA 0012 with a Mach at infinity equal to 2. and an angle of attack equal to 10 degrees.

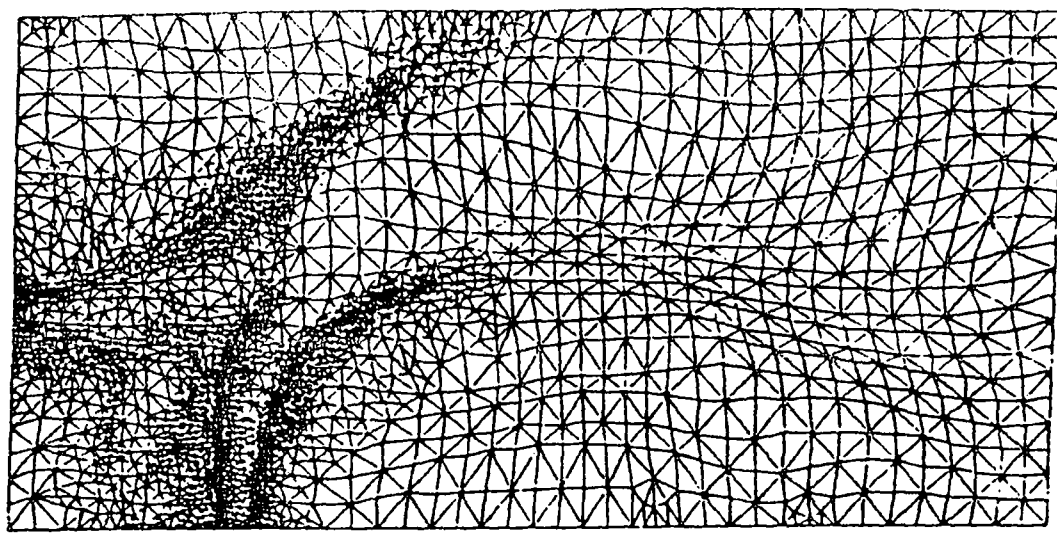


a) Mesh (2489 points).

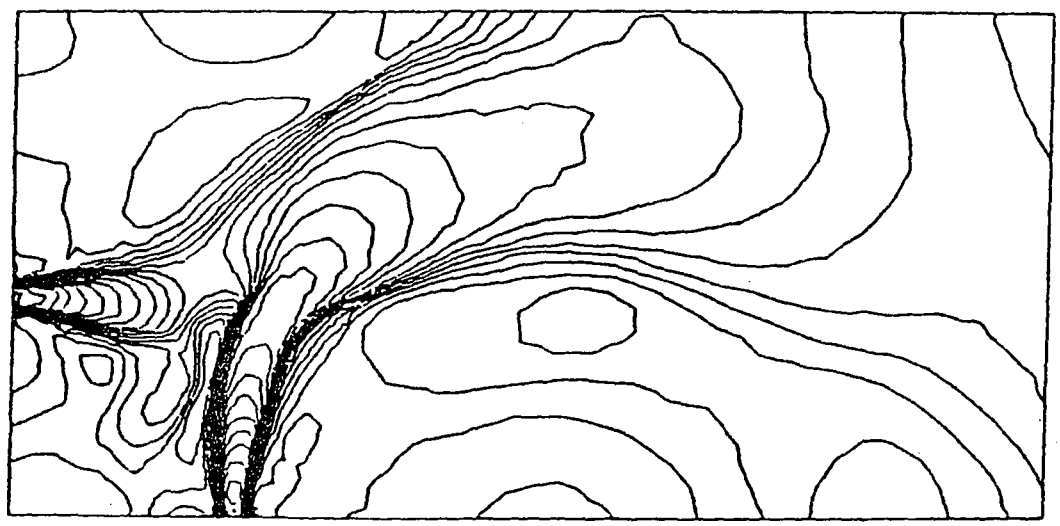


b) Isomach contours

FIGURE 17 : Same as Fig. 16, after enrichment and movement



a) Final mesh (2438 points)



a) Isomach contours

Figure 18 : Interacting jets

REFERENCES

[AND] W.K. ANDERSON, J.L. THOMAS, B. VAN LEER, A comparison of Finite volume Flux vector splitting for the Euler Equations, AIAA 23rd Aerospace Sciences Meeting, January 1985, Reno, Nevada.

[ANG1] F. ANGRAND, V. BILLEY, A. DERVIEUX, J. PERIAUX, C. POULETTY, B. STOUFFLET, 2-D and 3-D Euler flow calculations with a second-order accurate Galerkin F.E.M. AIAA Paper 85-1706.

[ANG2] F. ANGRAND, V. BILLEY, J. PERIAUX, C. POULETTY, J.P. ROSENBLUM, 2-D and 3-D Euler computations around lifting bodies on self adapted finite element meshes, Sixth Int. Symp. Finite Element in Flow Problems, Antibes (France),June 16-20, 1986

[ANG3] F. ANGRAND, V. BOULARD, A. DERVIEUX, J. PERIAUX, G. VIJAYASUNDARAM, Triangular Finite Element Methods for the Euler Equations, 6th International Conference on Computing Methods in Applied Sciences and Engineering, Glowinski R., Lions J.L., Eds., Norh Holland (1984).

[ANG4] F. ANGRAND, A. DERVIEUX, V. BOULARD, J. PERIAUX, G. VIJAYASUNDARAM, Transonic Euler Simulations by means of Finite Element Explicit Schemes, AIAA 6th Computational Fluid Dynamics Conference, July 13-15, 1983, Danvers, Mass., AIAA Paper 83-1924

[ANG5] F. ANGRAND, J. ERHEL, Vectorized Finite Element codes for compressible flows, Proceedings of "Finite Element in Flow Problem", Antibes (F), June 16-20 1986, to be published by Wiley and son

[ANG6] F. ANGRAND, P. LEYLAND, Schéma multigrille dynamique pour la simulation d'écoulements de fluides visqueux compressibles, INRIA, Research report no 659.

[ANG7] F. ANGRAND, P. LEYLAND, Une méthode Multigrille pour la résolution des équations de Navier-Stokes compressible en deux dimensions, INRIA Research Report 593 (1986)

[ANG8] F. ANGRAND, V. BILLEY, J. PERIAUX, J.P. ROSENBLUM, B. STOUFFLET, Implicit Euler calculations using a Galerkin Finite-Element approximation on adapted non-structured meshes. GAMM workshop, Rocquencourt, june 10-13, 1986, be published by Vieweg.

[BAB] K. BABA, M. TABATA, On a conservative upwind finite element scheme for convective diffusion equations, RAIRO, vol. 15, no 1, 3-25 (1981).

[BAI] M.J. BAINES, A.J. WATHEN, "Moving finite elements modelling of compressible flow", Reading University report (1985).

[BAK1] T.J. BAKER, A. JAMESON, N.P. WEATHERILL, Calculation of Inviscid Transonic Flow Over a Complete Aircraft. AIAA 24th Aerospace Sciences Meeting, January 1986, Reno, Nevada. AIAA Paper 86-01303

[BAK2] T.J. BAKER, A. JAMESON, R.E. VERMELAND, (1985) Three Dimensional Euler solutions with Grid Embedding. AIAA paper 85-0121.

[BAN] R. BANK, A. SHERMAN, "A multi-level iterative method for solving finite-element equations", Proceedings of the fifth symposium on reservoir simulation, pp. 117-126, Society of petroleum engineers of AIME, (1980).

[BEA] R. BEAM, and R. F. WARMING, "An Implicit Finite-Difference Algorithm for Hyperbolic Systems in Conservation-Law-Form", Journal of Computational Physics, Vol. 22, Sept. 1976, pp. 87-110.

[BEN] F. BENKHALDOUN, A. DERVIEUX, G. FERNANDEZ, H. GUILLARD, B. LARROUTUROU, "Some finite-element investigations of stiff combustion problems : mesh adaption and implicit time-stepping", Proceedings of the NATO advanced research workshop on Mathematical medelling in combustion and related topics, Brauner Schmidt-Lainé eds., to appear, (1987).

[BER] M.J. BERGER, A. JAMESON. (1984) An adaptive multigrid method for the Euler equations in 9th Int. Conf. on Num. Methods in Fluid dynamics, Lecture Notes on Physics, 218, 92-97.

[BIL1] V. BILLEY, (1984), Résolution des Equations d'Euler par des méthodes d'Eléments Finis ; Applications aux écoulements 3-D de l'Aérodynamique, Thésis, University of Paris VI.

[BIL2] V. BILLEY, J. PERIAUX, P. PERRIER, B. STOUFFLET, 2-D and 3-D Euler computations with Finite Elements Methods in Aerodynamics, International Conference on Hyperbolic Problems, Saint-Etienne, Jan. 13-17 (1986)

[BOH] K. BOHMER, P. W. HEMKER, and H. J. STETTER, The Defect Correction Approach, in : Defect Correction Methods. K. Böhmer, H. J. Stetter, eds., Comp. Suppl. 5, 1-32, Springer - Verlag, Wien. New York, 1984.

[BOR] M. BORREL, J.L. MONTAGNE, Numerical Study of a Non-Centered Scheme with Application to Aerodynamics, AIAA 18th Fluid Dynamics and Plasmadynamics and Lasers Conference, July 16-18, 1985, Cincinatti, AIAA Paper 85-1497

[BRA] A. BRANDT, S.F. MC CORMICK, J. RUGE, Algebraic multigrid (AMG) for space matrix equations, in Sparsity and its applications, (D.J. Evans Ed.), Cambridge University Press (1984).

[CHA] S.R. CHAKRAVARTHY, Relaxation Methods for Unfactored Implicit Upwind Schemes, AIAA 2nd Aerospace Sciences Meeting, January 9-12, 1984, Reno, Nevada, AIAA Paper 84-0165 (to appear in AIAA Journal)

[CHA] S.R. CHAKRAVARTHY, K.Y. SZEMA, An Euler Solver for Three Dimensional Supersonic Flows with Subsonic Pockets, AIAA 18th Fluid Dynamics and Plasmadynamics and Lasers Conference, July 16-18, 1985, Cincinnati, AIAA Paper 85-1703

[DAV] S.F. DAVIS, J.E. FLAHERTY, An adaptive finite-element method for initial boundary value problems for partial differential equations, SIAM J. Sci. Stat. Comp., 3, p. 6, (1982).

[DER1] A. DERVIEUX, (1985). Steady Euler simulation using unstructured meshes, von Karman Institute for Fluid Dynamics, Lecture Series 1985-04, Computational Fluid Dynamics, Lecture Series 1985-04, Computational Fluid Dynamics, March 25-29, 1985.

[DER2] A. DERVIEUX, J-A. DESIDERI, F. FEZOUI, B. PALMERIO, J.P. ROSENBLUM, B. STOUFFLET, Euler calculations by upwind finite element methods and adaptive mesh algorithms, GAMM workshop on the numerical Simulation of compressible Euler-flows, 10-13 Juin 1986. To appear in MECANICA COMPUTACIONAL, AMCA (Argentine).

[DER3] A. DERVIEUX, B. LARROUTUROU, B. PALMERIO, A review of mesh adaption methods (to appear).

[DES1] J-A. DESIDERI, Preliminary results on the iterative convergence of a class of implicit schemes, rapport INRIA No. 490, 1986.

[DES2] J-A. DESIDERI, Diagonalization of Implicit Upwind Schemes, in preparation.

[DES3] J-A. DESIDERI, E. HETTENA, Numerical simulation of hypersonic equilibrium-air reactive flow, INRIA Research Report no 716.

[DES4] J.A. DESIDERI, Real gases, Von Karman Institute Lecture Series 1988-05, March 7-11, 1988.

[DWY] H.A. DWYER, A discussion of some criteria for the use of adaptive gridding, AIAA paper 83-1932, (1983).

[EBE] A. EBERLE, K. MISEGADES. Euler Solution for a Complete Aircraft at and Supersonic Speed, 58th Meeting on the Fluid Dynamics Panel Symposium on Application of Computational Fluid Dynamics in Aeronautics (AGARD) April 7-10 1986, Aix en Provence, France.

[FEZ1] F. FEZOUI, Résolution des équations d'Euler par un schéma de van Leer en éléments finis, INRIA-Report 358 (1985).

[FEZ2] F. FEZOUI, V. SELMIN, H. STEVE, INRIA Report to appear.

[FEZ3] F. FEZOUI, B. STOUFFLET, A Class of Implicit Upwind Schemes for Euler Simulation with Unstructured Meshes, submitted to the J. of Comp. Phys.

[FEZ4] F. FEZOUI, B. STOUFFLET, J. PERIAUX, A. DERVIEUX, Implicit high-order upwind Finite-Element schemes for the Euler equations, Fourth Int. Symposium on Numerical Methods in Engineering (Atlanta, USA, March 24-28 1986), to be published by computational Mechanics Pub., Southampton(UK)

[FEZ5] L. FEZOUI, M-H. LALLEMAND, H. STEVE, A. DERVIEUX, New developments in Fast Euler Finite-Element solvers, 8th GAMM Conference, Louvain 1987, to be published by Vieweg.

[GNO] P.A. GNOFFO, A finite Volume adaptive grid algorithm applied to planetary entry flowfields, AIAA J., Vol. 21 , no 9 : 1249-1254 (1983).

[HEM] P. HEMKER, S.P. SPEKREIJSE. Multiple grid and Osher's scheme for the efficient solution of the steady Euler equations, Appl. Num. Math. 2 (1986), pp. 475-493.

[HIR1] C. HIRSCH, H. DECONINCK, A Multigrid Method for the Transonic Full Potential Equation Discretized with Finite Element on an Arbitrary Body Fitted Mesh, NASA Conference Publication 2202, 1981.

[HIR2] C.W. HIRT, A.A. AMSDEN, J.L. COOK. An arbitrary Lagrangian-Eulerian Computing method for all flow speeds, J. Comp. Phys. 14 (1974), 227-253.

[HUG] T.J.R. HUGHES, M. MALLET, L.P. FRANCA, New Finite Elements Methods for the compressible Euler and Navier-Stokes Equations, in 7th International Conference on Computing Methods in Applied Sciences and engineering, Dec. 9-13, 1985, Versailles (F), to be published by North Holland.

[HYM1] J.M. HYMAN, B. LARROUTUROU, Dynamic rezone methods for partial differential equations in one space dimension, to appear.

[HYM] J.M. HYMAN, M.J. NAUGHTON, Static rezone methods for tensor-product grids, Los Alamos National Laboratory Report LA-UR-83-3245, (1983).

[JAM1] A. JAMESON, Numerical solution of the Euler equations for compressible inviscid fluids, Numerical methods for the Euler equations of Fluid Dynamics, F.Angrand et al. Eds., SIAM Philadelphia (1985)

[JAM2] A. JAMESON, T.J. BAKER, N.P. WEATHERILL, Calculation of inviscid transonic flow over a complete aircraft, AIAA Paper 86-0103 (1986)

[LAL1] M-H. LALLEMAND, Thesis, march 1988, Marseille (France).

[LAL2] M-H. LALLEMAND, Etude de schémas Runge-Kutta à 4 pas pour la résolution multigrille des équations d'Euler 2-D. INRIA Research Report (in preparation).

[LAL3] M-H. LALLEMAND, A. DERVIEUX, A multigrid finite element method for solving the two dimensional Euler equations, Third Copper Mountain Conference on Multigrid Methods, april 6-10, 1987, Copper Mountain, Colorado, U.S.A.

[LAL4] M-H. LALLEMAND, F. FEZOUI, E. PEREZ, Un schéma multigrille en Eléments Finis décentré pour les équations d'Euler, INRIA Research Report 602 (1987).

[LAL5] M-H. LALLEMAND, and H. STEVE, private communication.

[LEC] M.P. LECLERC, Communication to "Colloque National d'Analyse Numérique ",mai 1988, Evian (France).

[LEE1] B. VAN LEER, Computational Methods for Ideal Compressible Flow, Von Karman Institute for Fluid Dynamics. Lectures series 1983-04 Computational Fluid Dynamics, March 7-11 1983.

[LEE2] B. VAN LEER, Flux-Vector splitting for the Euler equations, Lecture Notes in Physics, Vol. 170, 570-512 (1982).

[LEE3] B. VAN LEER, W.A. MULDER. Relaxation methods for hyperbolic equations, Proceedings of the INRIA workshop on "Numerical methods for compressible inviscid fluids", dec. 7-9 1983, Numerical Methods for the Euler Equations of Fluid Dynamics, Angrand F. et al Eds., SIAM (1985)

[LOH1] R. LOHNER, K. MORGAN, Unstructured Multigrid methods, Second European Conference on Multigrid Methods, Koln (RFA), October 1-4, 1985

[LOH2] R. LOHNER, K. MORGAN, J. PERAIRE, O.C. ZIENKIEWICZ, L. KONG, Finite Element Methods for Compressible Flows, Proc. of the ICFD Conference on Numerical Methods for Fluid Dynamics , April 1-4 1985, Reading (UK), Numerical Methods for Fluid Dynamics, II, K.W. Morton, M.J. Baines Eds., Clarendon, Oxford (1986)

[MC] R. W. MacCORMACK, Current Status of Numerical Solutions of the Navier-Stokes Equations, AIAA Paper 85-0032. 1985.

[MAV] D. MAVRIPLIS, A. JAMESON, Multigrid solution of the two-dimensional Euler equations on unstructured triangular meshes, AIAA Paper 87-0353.

[MIL1] K. MILLER, Moving finite elements II, SIAM J. Numer. Anal., Vol. 18, 1033-1057, (1981).

[MIL2] K. MILLER, R.N. MILLER, Moving finite elements I, SIAM J. Numer. Anal., Vol. 18, 1019-1032, (1981).

[MOR1] K. MORGAN, J.R. JONES, J. PERAIRE, M. VAHDATI, R. LOHNER, Finite Element FCT for the Euler and Navier-Stokes equations, Sixth Int. Symposium "Finite Element Methods in Flow Problems", Antibes (F), June 16-20, 1986

[MOR2] K. MORGAN, J. PERAIRE, Finite-element methods for compressible flows, Computational fluid dynamics, Lecture series 1987-04, Von Karman Institute for fluid dynamics, (1987).

[MUL] W. A. MULDER, Multigrid Relaxation for the Euler Equations, J. Comp. Phys. 60 (1985), pp. 235-252.

[NAK] K. NAKAHASHI, G.S. DIEWERT, (1985) A practical adaptive grid method for complex fluid flow problems in 9th Int. Conf. on Num. Methods in Fluid Dynamics, Lecture Notes in Physics, 218, 422-426.

[ORT] J. M. ORTEGA, and W. C. RHEINBOLDT, "Iterative Solution of Nonlinear Equations in Several Variables", Academic Press, New York, 1970.

[OSH1] S. OSHER, S. CHAKRAVARTHY, Upwind difference schemes for hyperrbolic systems of conservation laws, Math. Computation, April 1982.

[OSH2] S. OSHER, S. CHAKRAVARTHY, Upwind schemes and boundary conditions with applications to Euler equations in general geometries, J. of Comp. Physics, Vol. 50, 3, 447-81 (1983).

[PAL1] PALMERIO B., Self-adaptive F.E.M. algorithms for the Euler equations, IN-

RIA Report 338 (1984).

[PAL2] PALMERIO B., A two dimensional F.E.M. adaptive moving node method for steady Euler flow simulation, submitted to Computer Methods in Applies Mechanics and Engineering.

[PAL3] B. PALMERIO, V. BILLEY, A. DERVIEUX, J. PERIAUX, Self Adaptive Mesh Refinements and Finite Element Methods for Solving the Euler Equations, Proc. of the ICFD Conference on Numerical Methods for Fluid Dynamics, April 1-4. 1985, Reading (UK), Numerical Methods for Fluid Dynamics, II, K.W.Morton, M.J.Baines Eds., Clarendon, Oxford (1986)

[PAL4] B. PALMERIO, A. DERVIEUX, Application of a F.E.M. moving node adaptive method to accurate shock capturing, in Numerical grid generation in Computational Fluid Dynamics, Hauser J., Taylor C., eds. Pineridge Press Swansea 1986.

[PAL5] B. PALMERIO, Recent developments in adaptive finite-element calculations of compresible flows, Conference on "Automated Mesh Generation and Adaption", oct. 1-2, 1987, Grenoble (France), Math. Prepub. 163, Nice University.

[PER1] E. PEREZ, Finite Element and Multigrid solution of the two-dimensional Euler equations on a non-structured mesh, INRIA Report 442 (1985)

[PER2] E. PEREZ, J. PERIAUX, J.P. ROSENBLUM, B. STOUFFLET, A. DERVIEUX, M-H. LALLEMAND, Adaptive Full-Multigrid Finite Element Methods for solving the two-dimensional Euler equations, I.C. 10 N.M.F.D, June 1986, Beijing, to be published by Springer

[RAK] J.V. RAKICH, H.E. BAILEY, C. PARK, Computation of Non-equilibrium Three-Dimensional Inviscid Flow over Blunt-Nosed Bodies Flying at Supersonic Speeds, AIAA Paper 75-835 (1975)

[ROS] J.P. ROSENBLUM, Thesis, in preparation.

[SAA] Y. SAAD, M.H. SCHULTZ, : GMRES : A Generalized Minimal Residual algorithm for solving nonsymmetric linear systems, YALE Univ. Research Report YALEU/DCS/RR-254 (1983)

[SCH] W. SCHRODER, D. HANEL, An unfactored implicit scheme with multigrid acceleration for the solution of the Navier-Stokes equations, Computer and Fluids vol. 15, no 3, pp. 313-336, 1987.

[SEL] V. SELMIN, Finite Element Solution of Hyperbolic equations, II, Two Dimen-

sional case, INRIA Research Report 708. 1987.

[STE1] J.L. STEGER, R.F. WARMING, Flux Vector splitting for the inviscid gas dynamic equations with applications to finite difference methods, Journal Comp. Physics, vol. 40, no. 2, pp 263-293, 1981.

[STE2] J. L. STEGER, Coefficient Matrices for Implicit Finite Difference Solution of the Inviscid Fluid Conservation Law Equations, Computer Methods in Applied Mechanics and Engineering 13 (1978). pp. 175-188.

[STE3] J. L. STEGER, Implicit Finite-Difference Simulation of Flow about Arbitrary Geometries with Application to Airfoils, AIAA Paper 77-663 (1977).

[STE4] J.L. STEGER, R.F. WARMING, Flux Vector splitting for the inviscid gas dynamic equations with applications to finite difference methods, Journal Comp. Physics, vol.40, No. 2, pp 263-293, 1981.

[STEV1] H. STEVE, Thesis, in preparation.

[STEV2] H. STEVE, "Méthodes Implicites Efficaces pour la Résolution des Equations d'Euler en Eléments Finis", INRIA Report no 779 Decembre 1987.

[STO1] B. STOUFFLET, Implicit Finite Element Methods for the Euler Equations. Proceedings of the INRIA Workshop on "Numerical Methods for Compressible Inviscid Fluids, 7-9 dec. 1983, Rocquencourt (France), Numerical Methods for the Euler Equations of Fluid Dynamics, Angrand F. et al Eds., SIAM (1985)

[STO2] B. STOUFFLET, "Résolution Numérique des Equations d'Euler des Fluides Parfaits Compressibles par des Schémas Implicites en Eléments Finis", thesis, Université Pierre et Marie Curie. Paris VI, 1984.

[STO3] B. STOUFFLET, J. PERIAUX, F. FEZOUI, A. DERVIEUX, "Numerical Simulation of 3-D Hypersonic Euler Flows Around Space Vehicles Using Adapted Finite Elements", AIAA-87-0560. AIAA 25th Aerospace Sciences Meeting, January 12-15, 1987/Reno, Nevada.

[STR] G. STRANG, "Linear Algebra and its Applications", Second Edition, Academic Press, New York, 1980.

[THO] J. L. THOMAS, B. van LEER, and R. W. WALTERS, "Implicit Flux-Split Schemes for the Equations", AIAA Paper 85-1680, 1985.

[TUR] E. TURKEL, B. VAN LEER, Flux vector splitting and Runge-Kutta methods for the Euler equations, ICASE Report 84-27. June 1984.

[USA] W.J. USAB, E.H. MURMAN, (1983) Embedded Mesh Solutions of the Euler Equations using a Multiple-Grid Method in 6th AIAA Conference on Computational Fluids Dynamics (Danvers, Mass. USA), July 13-15 1983.

[VAN] P. VANCAMBERG, Experimental and Theoretical Tests for the Prediction of Aerodynamical Moments of HERMES in Hypersonic Flight, ICAS Paper ICAS 86.2.10.1 (1986)

[VAR] R. S. VARGA, "Matrix Iterative Analysis", Prentice Hall, Inc., Enlewoods Cliffs, N. J., 1962.

[VIJ] G. VIJAYASUNDARAM, Transonic flow simulations using an upstream centered scheme of Godunov in Finite Element, J.C.P., vol. 63, 416-433 (1986).

[WAR] R.F. WARMING, B.M. BEAM, B.J. HYETT, Diagonalizaton and simultaneously symmetrization of the Gas-Dynamic Matrices, Mathematics of Computation, Vol. 29, 132 (1975), pp 1037-1045.

[ZIE1] ZIENKIEWICZ O., (1971) The Finite Element Method in Engineering Science, McGraw-Hill, London.

[ZIE2] ZIENKIEWICZ O.C., Adaptive mesh refinement in flow problems, Sixth Int. Symposium "Finite Element Methods in Flow Problems", Antibes (F), June 16-20, 1986

ISSN 0249-6399